

年 報

第 2 号

2005年2月

香川大学総合情報基盤センター

目 次

平成16年度年報に寄せて	乗松尋道 ...	1
研究・自己紹介		
教育工学研究：IT活用による新しい教育・学習環境を目指して	林 敏浩 ...	5
今更ですが、LINUX&C 言語プログラミングも捨てがたい！	今井慈郎 ...	10
複雑な生体システムを生命情報学の切り口で探求	岩間久和 ...	12
生命現象を分子の3次元構造から解明する	神鳥成弘 ...	13
縁ありて基盤センターへ	上原正宏 ...	14
X線結晶構造学に足をふみいれて	吉田裕美 ...	15
生物の進化とゲノムの進化	竹崎直子 ...	16
キャンパス情報システムにおけるヒューマンネットワークの形成支援 ...	堀 幸雄 ...	17
第1回総合情報基盤センター先端研究セミナー開催報告	吉田裕美, 神鳥成弘 ...	19
教育		
分子生物学（医学部医学科2年生後期）講義の概要紹介	竹崎直子, 岩間久和, 神鳥成弘, 吉田裕美 ...	23
教育活動	上原正宏 ...	24
集中講義「知的CAI」	林 敏浩 ...	26
高校生へのセンター紹介	竹崎直子, 岩間久和 ...	27
キャンパスウォークにおけるセンター紹介	堀 幸雄 ...	28
情報基盤		
香川大学のネットワーク構成	林 敏浩, 河内一芳, 瀬野芳孝, 曾根計俊 ...	31
総合情報基盤センター業務・サービス	河内一芳, 川田延枝, 池田裕子 ...	32
幸町分室の紹介	堀 幸雄, 瀬野芳孝, 曾根計俊, 高橋多恵 ...	34
活動報告		
アタックレポートの集計	青木昌三, 瀬野芳孝, 曾根計俊, 高橋多恵 ...	39
ウィルスチェックサーバによるアラートメールの集計	青木昌三, 瀬野芳孝, 曾根計俊, 高橋多恵 ...	42
Apache James の設定と動作検証() - SMTP サーバとしての基本的な設定 -	青木昌三 ...	47
Apache James の設定と動作検証() - 拡張機能の設定 -	青木昌三 ...	82

Apache James の設定と動作検証() - Matcher, Maillet の作成と組み込み -	青木昌三 ...	112
教育用 PC システムの保守・更新作業について	黒田 勉, 香川考司, 瀬野芳孝, 曾根計俊 ...	126
ウェブページアクセスログ解析	中村邦彦 ...	129
香川県における医療 IT への取り組みと今後の課題	原 量宏 ...	150
医学部池戸地区基盤センター無線 LAN 接続システム・研究支援システム ...	木村正司 ...	151
認証 DHCP サービスの利用状況	最所圭三, 瀬野芳孝, 曾根計俊 ...	152
業務報告		
利用申請一覧		161
計算機利用統計		168
センター利用状況		172

平成 16 年度年報によせて

センター長 乗松尋道

平成 15 年 10 月に行われた香川大学と香川医科大学との統合を期に総合情報基盤センターが設置されました。この新たなセンターは旧香川大学の情報処理センターと旧香川医科大学の情報メディアセンターそれぞれに張ってあったネットワークを単に接続したのではなく、学内共同研究施設として専任教員の配置が計画され、高いレベルの研究拠点として位置づけられました。センターは情報基盤システム部門、情報応用システム部門、情報教育システム部門の 3 部門からなり、教授 2 名、助教授 5 名(内 1 名は教授の振りかえ)、助手 1 名の計 8 名が全国公募で選考され、先日の 2 月 1 日に最後の教員が赴任されました。

センターの業務の一つは学内の 4 つのキャンパスを結んでいるネットワーク幹線を維持管理し、それをさらに地域貢献、国内外への情報発信に役立て、新たな通信手段の開発に努力することです。2 番目の業務は情報教育ですが、PC の単なる使用方法に関する教育ではなく、学部学生、大学院学生がそれぞれの専門領域で高度な情報技術を応用するためのものであり、またこのような情報教育を実践できる人材育成も含まれます。本センターの重要な業務の最後は、生命情報科学分野で基礎を築き、学内、学外との共同研究を積極的に推進し、ゲノム情報科学、分子構造機能学、応用解析学、生命認知工学、教育工学などの分野で拠点形成を行うことにあります。

本学の情報化はセンターばかりでなく、事務部局、学務関連部局、附属図書館、附属病院医療情報部などがそれぞれ独立して進めてきましたが、今後は統一した推進機構を構築し、またそれぞれの部局の支線部分も発展させていく必要があります。この点につきましては全学ネットワークのセキュリティポリシー策定と共に、企画担当上杉正幸理事を中心に鋭意検討されているところです。今年度はこのように教員が整備され、組織の確立に努めて参りましたが、今後はさらにセンターの研究活動を強化し、その業務内容を検討していかなばならないと考えておりますので、学内の各学部、研究科、各部局の皆様のご協力、ご支援を宜しくお願い致します。

研究・自己紹介

教育工学研究：IT 活用による新しい教育・学習環境を目指して

所属：総合情報基盤センター

氏名：林 敏浩

1. はじめに

教育工学は工学と教育学の境界領域の複合研究であり、米国コミュニケーション工学会の定義によれば「教育工学とは、学習の過程と資源についての設計、開発、運用、管理、ならびに評価に関する理論と実践である」と言われている。また、ソフトウェア開発として立場からは、著者がこれまで行ってきた研究は CAI (Computer Assisted Instruction) システムあるいは教育システム開発になる。なお、本分野を中心的に扱う学会としては、日本教育工学会、教育システム情報学会などがある。

次に、CAI システム研究の観点から簡単にこの分野の歴史を紹介する。CAI システム研究は比較的古くから行われており、1940 年代の米軍による魚雷発射訓練システム SAGE システムが知られている。世界最初の電子計算機 ENIAC が開発されたころの話である。しかし、最初の CAI システム開発は、1958 年にハーバード大学の Skinner 教授によるものと言われている。1960 年代になると、旧ソ連の人工衛星スプートニク打ち上げにより「スプートニクショック」を受けたアメリカが CAI 研究に対して大量の資金投下を行い、CAI 研究分野は発展期を迎え様々な CAI システムや関連ソフトウェアが提案・開発された。しかし、当時の CAI システム（伝統的 CAI システムと呼ばれる）は高価だが機能的に貧弱な計算機上に開発され、また、開発技術も未熟であった。このため、1970 年代になると伝統的 CAI システムの融通性のない機械的な教授は「高価な電子紙芝居」、「電子ページめくり機」というように大きな批判を受けた。その後、人工知能技術を取り入れた知的 CAI システムに研究パラダイムが移行した。さらに 1990 年代になると安価であるが高機能な計算機を背景としてマルチメディア技術、ネットワーク技術を活用した様々な教育システム（近年は CAI の言葉をあまり使われない）が提案・開発されるようになった。近年はネットワーク技術を中核に IT 活用による新しい教育・学習環境の提供として、教育への WWW 活用、遠隔教育、e-Learning などの研究が盛んになってきている。

本稿では、IT 活用による新しい教育・学習環境の構築・運用さらにはそれらを用いた教育実践として、著者が前任の佐賀大学で近年行った研究（共同研究を含む）に関して紹介する。

2. WWW を利用した演習型講義支援

近年、高度情報化社会に対応できる人材育成のために高等教育機関での情報教育の重要性が高まり、講義カリキュラムに情報教育科目が急速に導入されている。特に演習科目では PC などを利用した操作方法などの教授・学習が行われる。著者は情報教育の演習講義を対象に、学生が主体的に演習できる WWW 教材の開発を行い、それに基づき平成 10 年度より実際に演習講義を実践した。方法論としては、演習講義の流れとそれに必要な WWW 教材をモデル化し、それらに基づき情報教育の演習講義用の WWW 教材を作成し実際の演習講義の中で利用した。

演習講義の流れは、教員による座学と学生による演習に分類し、後者はさらに、模倣による知識・技能形成過程、試行錯誤による知識・技能洗練過程、問題解決による知識・技能確認過程の 3 段階に分類した。また、これに教員による評価の過程まで含めて講義をモデル化

した。このモデルに基づき、必要な WWW 教材の構造として導入教材、演示教材、評価教材、演習成績の 4 種類に限定した。さらにこの教材の実装として、学習コンテンツページ（導入・演示教材）、演習課題ページ（評価教材）、課題評価ページ（演習成績）を作成した。図1に作成した WWW ページ (PowerPoint の使い方) を示す。

毎年度、最終演習講義の後に質問紙により作成した WWW ページの有効性を調査したところ、学習が主体的に演習を進めていくための資料として活用されていることが分った。なお、本研究の詳細に関しては文献[1]を参照されたい。

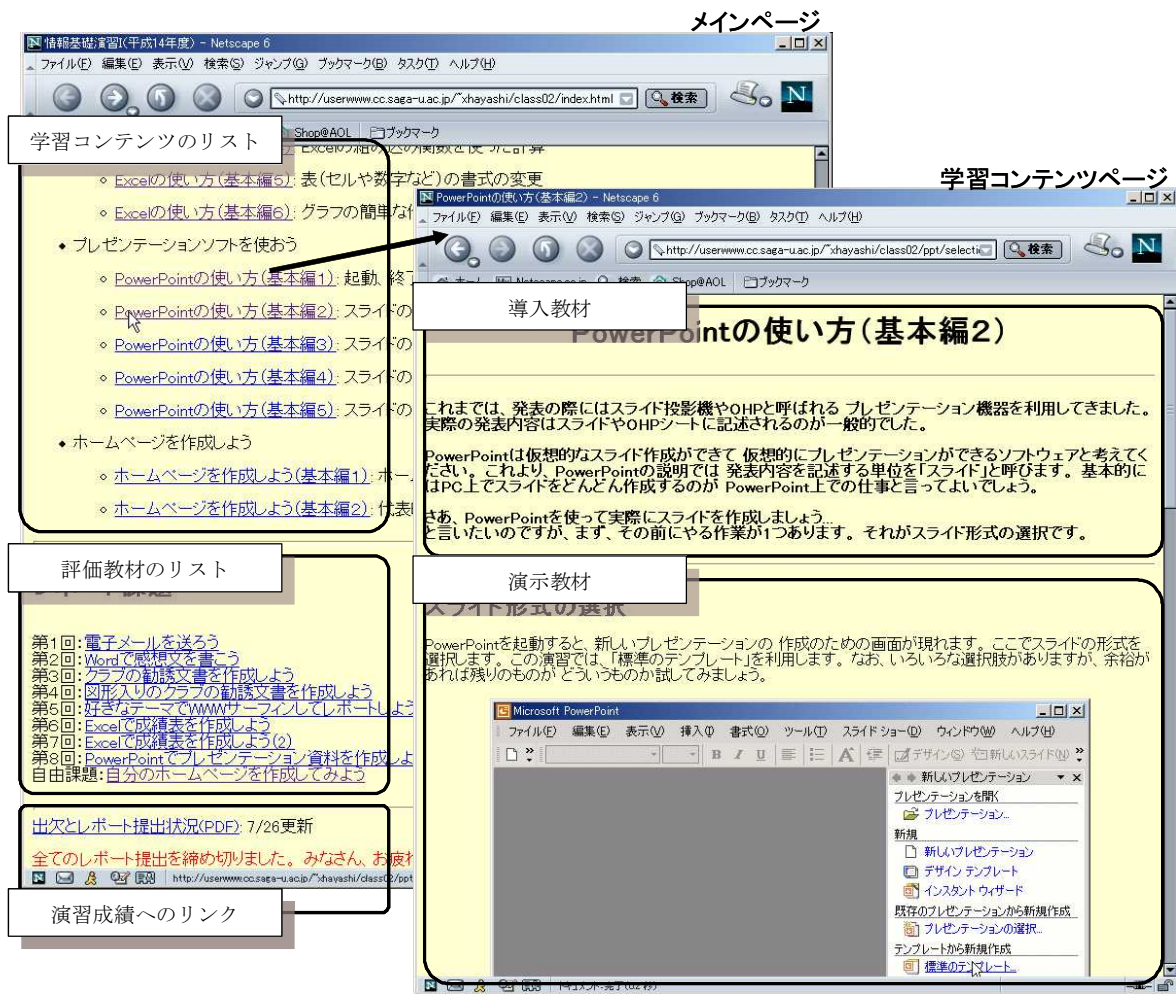


図1: 情報教育の演習講義を支援する WWW ページ

3. 超高速ネットワークを利用した遠隔講義

近年、情報ネットワークの発展に伴い多くの遠隔教育の研究や教育実践が行われている。特に、リアルタイム型の遠隔教育は、その実施規模により、遠隔講義（遠隔授業、交流授業を含む）、遠隔セミナー、e-Learning 型の個別学習に大別できる。受講者数が多い遠隔講義の形態は板書を用いた講義を遠隔化したものから、OHP やプレゼンテーションソフトなどを活用・併用した講義まで多様である。特に、本研究では、超高速ネットワークを用いた前者の板書を用いた遠隔講義の設計と運用方法について検討し、実際の教育実践を行った。

特に教育実践に関しては、佐賀大学と徳島大学を研究用超高速ネットワーク JGN (Japan Gigabit Network) で接続し、板書をベースにした遠隔講義を行った。図2に遠隔

講義の構成を示す。図中の CAM は DV カメラ、PRJ はビデオプロジェクタ、AMP は AV アンプを表す。

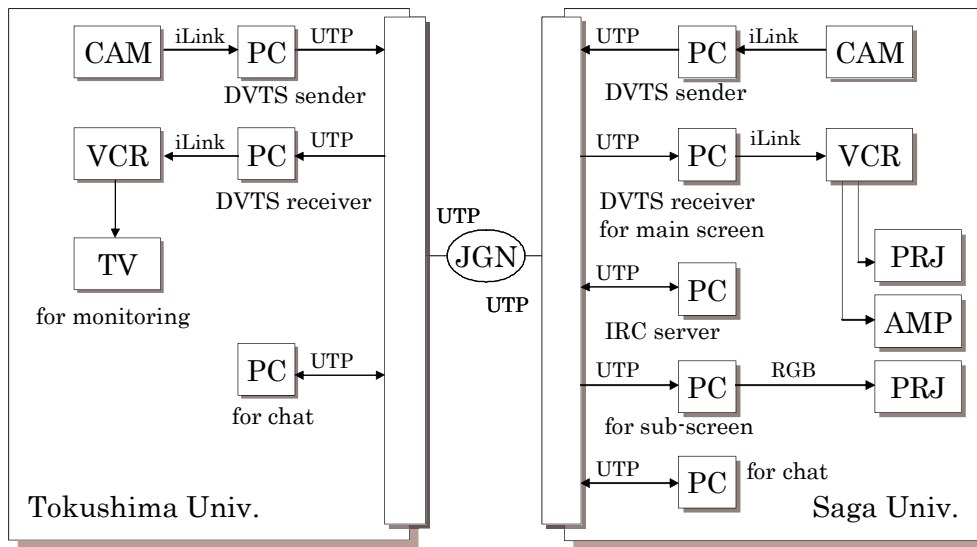


図 2 : JGN を利用した遠隔講義の構成

この実践では徳島大学から教員が佐賀大学の講義室にいる学生に講義を配信する形態を取った。また、図 3 にその実践時のスナップショットを示す。この研究を通して得られた設計と運用上の知見に関しては文献[2]を参照されたい。



図 3 : 遠隔講義の様子 (佐賀大学側)

4. 在宅での予習支援を指向したブレンディッドラーニング

近年、e-Learning の導入が企業を中心に増加している。また、大学などの高等教育機関への e-Learning 導入事例も講義の質や量の向上、新しい教育形態の提供、大学宣伝の一方法など多様な理由により増えている。著者は、佐賀大学理工学部の 2 つの専門講義科目に対し e-Learning

を導入し教育実践を行った。本実践では、対面授業と e-Learning の融合形態であるブレンディッドラーニングに着目した。本教育実践におけるブレンディッドラーニングは、在宅などでの予習を e-learning で提供し学生の自習を支援する。さらに、大学で実施する対面講義では、学生の応用力を向上させるために予習による前提知識に基づき演習形態の講義を展開する。特に、本研究では教授内容の質と量を向上した講義の実現を目指した。

本教育実践では、予習用 e-Learning 教材として 1 回の視聴時間が 10 数分程度のものを作成した。これは、何度も教材を見直して予習を深めること、学生が教材を途中で止めてノートを取ること、予習後のチェック課題の時間を確保すること、予習は長くても 2 時間程度で終わることを想定して、短めの e-learning 教材を作成した。

提供すべき予習教材のベースをマイクロソフト社の PowerPoint スライドとした。さらに PowerPoint の拡張ツールとして無償でマイクロソフト社から提供されている日本語版 Producer2002 とフリーウェアの録音ツールを用いて、説明用音声と PowerPoint スライドを統合した e-learning 教材を作成した。図 4 に本実践で作成した e-Learning 教材のスナップショットを示す。また、図 5 は e-Learning 教材をアクセスするための WWW ページを示す。本研究の詳細に関しては文献[3]などを参照されたい。

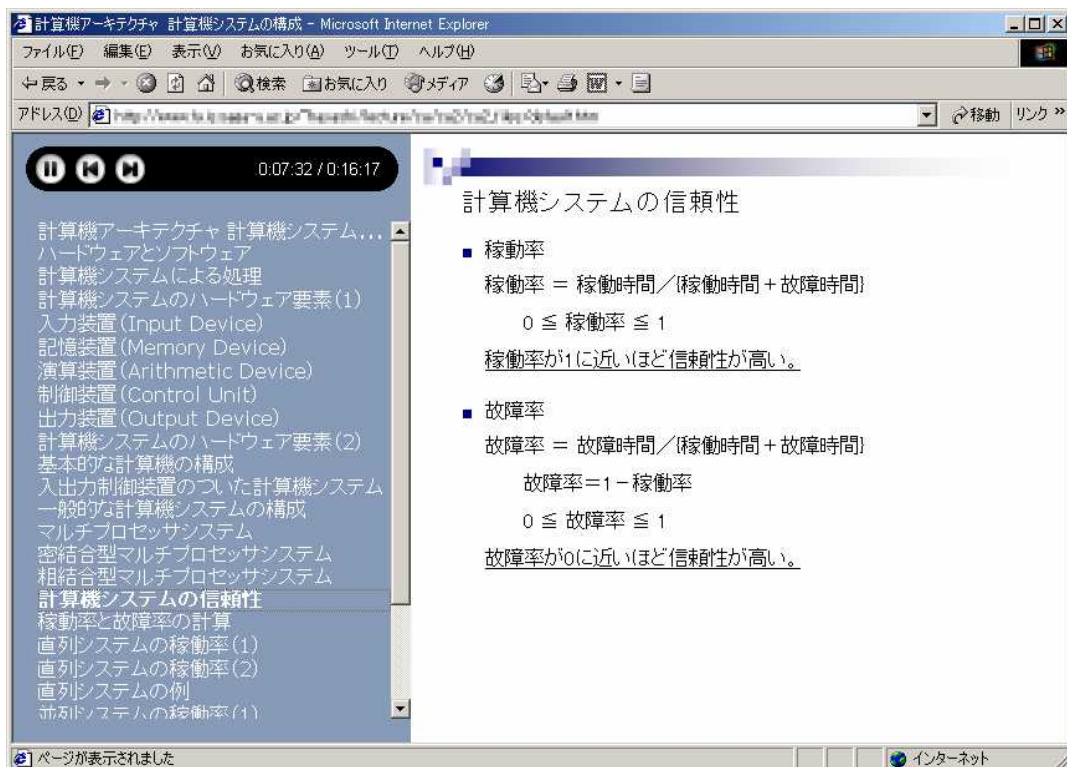


図 4：予習用 e-Learning 教材

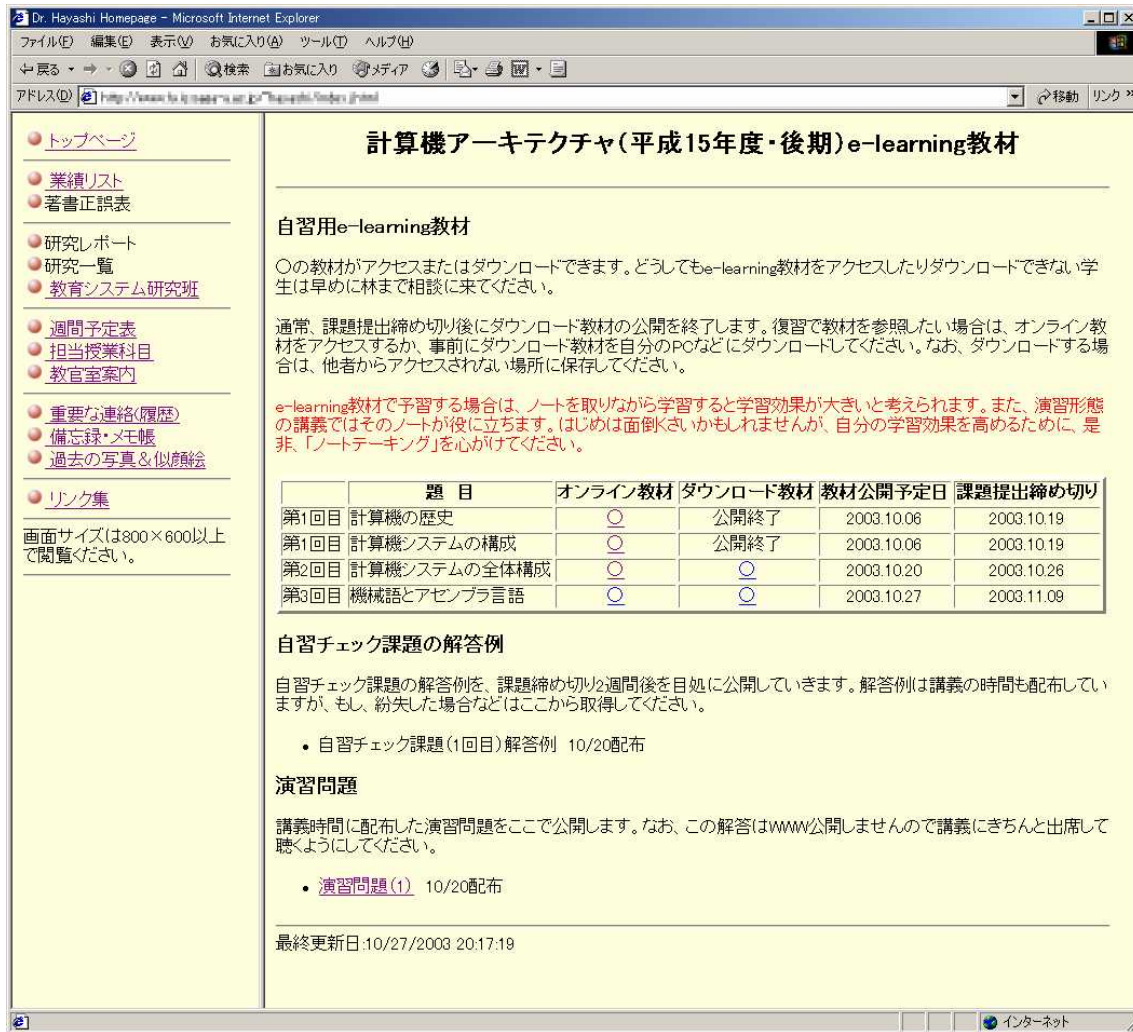


図5：e-Learning教材ページ

5. 最後に

本稿では、著者がこれまでに実施した、教育へのWWW応用、遠隔教育、e-Learningに関する研究を紹介した。今後も香川大学でこれらの研究を継続する予定である。興味のある方は連絡をいただくと幸いです(連絡先: hayashi@eng.kagawa-u.ac.jp)。

参考文献

- [1] 林敏浩, 林田行雄: "WWW教材を用いた情報教育のための演習講義の設計と実践", 教育システム情報学会誌, Vol.18, No.3/4, pp.377-386(2001).
- [2] 林敏浩, 渡辺健次, 大谷誠, 田中久治, 岡崎泰久, 林田行雄, 近藤弘樹: "高精細メディア機器と超高速ネットワークを用いた遠隔環境での板書型講義の設計と運用", 教育システム情報学会誌, 2005年掲載予定(2005).
- [3] 林敏浩, 林田行雄: "ブレンディッドラーニング型講義の設計と運用", 電気学会教育フロンティア研究会, FIE-04-10, pp.46-49(2004).

今更ですが，LINUX & C 言語プログラミングも捨てがたい！

所属： 総合情報基盤センター（情報基盤システム部門）

氏名： 今井慈郎

ここ数年，多くの先生方の指導もあり，JAVA プログラミング言語を用いた GUI プログラミング&ネットワークプログラミングをベースにした教育研究に「はまって」おりました(図1，図2を参照)．特に，遠隔監視・制御についてのテーマはクライアントに携帯電話を利用するなど，学生さんにとっても親しみ易いという傾向もあって，熱心に JAVA プログラミングに取り組んでくれていました．ところが，ある学生さんが，ネットワークカメラを LINUX マシン(OS を LINUX とするパソコン)と接続し，C 言語を用いて遠隔制御用のプログラムを記述するという研究を始めたので，こちら也大いに影響されました．彼には文才があり，しかもデザインなどにも興味があることから，研究室のロゴ(図3参照)まで作ってくれました．実は，インターンシップで貴重な経験をして，LINUX&C という少々オーソドックスな環境での研究環境や語り合える仲間を求めて，われわれの研究室の一員になってくれたという訳です．一緒に仕事ができる仲間がいるということは互いの年齢とは無関係ですし，楽しいことも多いと思います．総合情報基盤センターへの参加をきっかけに，これまで以上に教育研究の仲間を増やしていけるよう頑張りたいと思います．

以下は，現在，学生さんと一緒に取り組んでいる研究テーマです．それぞれ，担当学生との分担があるのもあって，ある時は教えられ，互いに刺激しあって結構楽しく取り組んでいます．

1) 「計算機の内部構造や動作を可視化するビジュアルシミュレータの開発」

計算機の内部動作を1命令ごとのステップ動作あるいは連続動作させることで，計算機内部の構造や振舞いを視覚的にも理解させることを目的に開発した自習可能な E-LEARNING 教材です．アセンブリ言語プログラミング演習にも使え，遠隔教育環境での使用を想定して教師・学生間での情報の遣り取りを支援するメールハンドラを組み込んでいます．JAVA プログラミングを用いて実現し，起動時に自動判断して，JAVA のアプレットとしても，スタンドローンアプリとしても動作します．授業でも利用しています．また，複数の大学や教育機関(海外も含む)から使用依頼も寄せられています(図1を参照)．

2) 「遠隔監視・制御システムの開発」

クライアントとして，モバイル PC や PDA(個人情報端末)，あるいは携帯電話(現時点ではドコモなど特定機種でのみ動作可能)を利用し，ネットワークカメラからの画像取得や，OPEN PLANET 技術を利用して電気回路を制御するなどの，遠隔監視，遠隔制御を実現しています．基本コンセプトは，「JAVA プログラミングでどこまで行けるか」であり，サーバ機能はもちろん，多種のクライアントにも対応できるようなシステム作りを目指しています．このテーマについて，学外の方(企業を含む)と共同研究しています．学生さんに最近，人気のあるテーマの1つで，携帯電話アプリの開発などに関連して，関連する企業の方にも支援をいただいています(図2を参照)．

3) 「ハードウェア製作・ソフトウェア作成を含む情報システム工学実験環境の構成」

情報システムでは，ハードウェアおよびソフトウェアへの理解と技術習得は重要な教育テーマとなります．一般に，windows ベースの教材が少ないのですが，ここでは，unix リテラシーの修得を目指し，「LINUX ベースの実験環境」を構成し，維持管理することを目的に

しています。従って、簡単なデジタル回路の製作およびそのデバッグ、製作したデジタル回路を linux ベースの pc に接続し動作させるためのデバイスドライバの作成など幅広いテーマを効率よく扱おうとしています。また、unix リテラシー修得の一環として、ネットワーク環境を理解し、ネットワークユーティリティの利用もテーマの1つとなります。

4) 「C 言語による WEB サーバの作成と監視カメラ制御への応用」

これも LINUX マシンをベースに、C 言語による WEB サーバの作成とそれを利用した監視カメラの制御および画像情報のデータベース化などを研究テーマとしたプロジェクトです。JAVA でのプログラミングが中心だったので、久しぶりという感じですが、やはり C 言語プログラミングは小回りが効くと結構、楽しんでます。企業の方から多くの支援を得てスタートした新プロジェクトです。これまでフリーウェアとは言え、開発済みのソフトウェアを利用するという状況が多かったのですが、このプロジェクトは WEB サーバを作成し、それにカメラ制御に関する手続きなどを付加しては動作をテストするという大変オーソドックスな手法を採用しています(但し、LINUX についてはエンドユーザですが)。将来は、組み込みソフトウェア開発(リアルタイム処理を含む)もカバーできればと計画しています。

現状では、上記のような内容となっておりますが、今後はネットワークやデータベースを教育研究テーマにした題材も検討したいと考えております。また、昨今、情報セキュリティが大きな話題となっております。総合情報基盤センターに参加させていただいたことを1つの契機として、「情報セキュリティ」問題の勉強も始めたいと考えております。

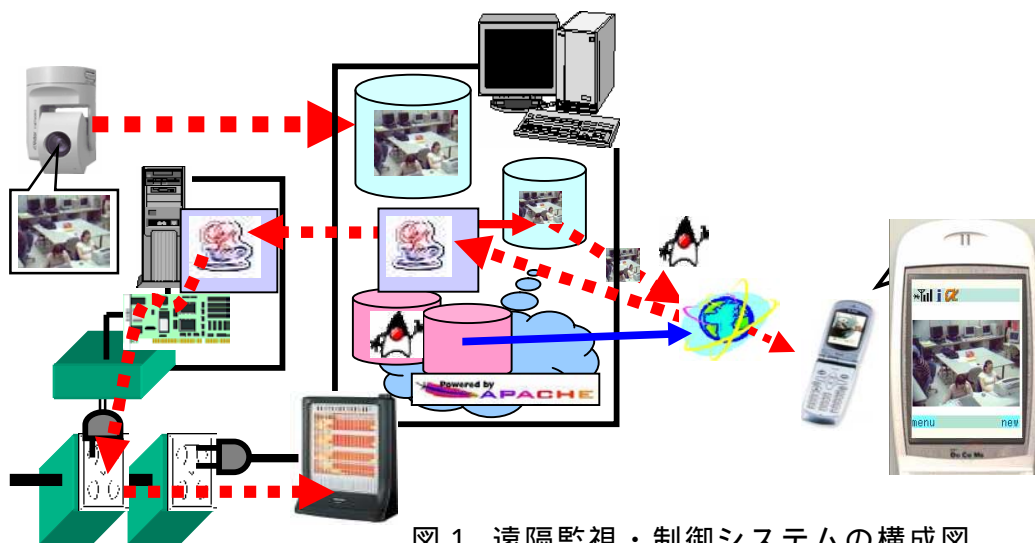


図1 遠隔監視・制御システムの構成図

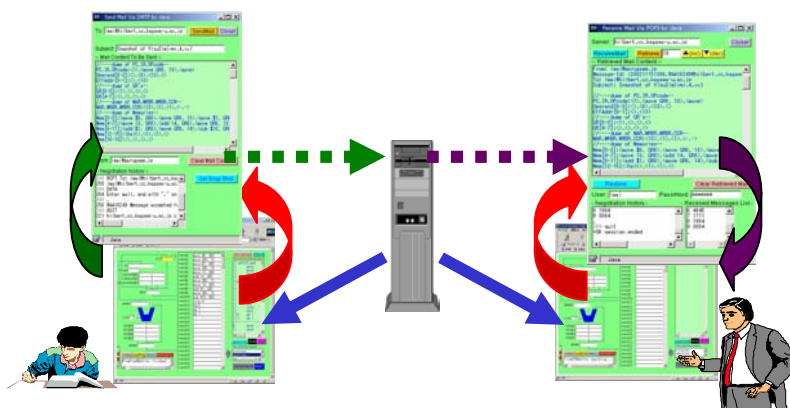


図2 ビジュアルシミュレータのネットワーク利用環境



図3 研究室のロゴ

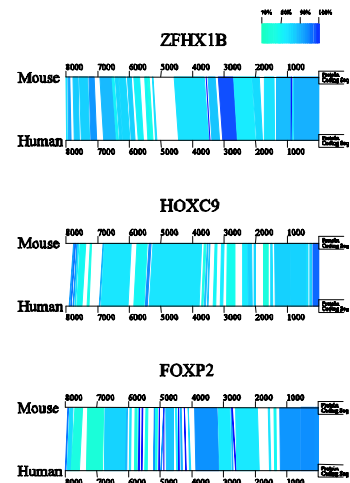
複雑な生体システムを生命情報学の切り口で探求

所属： 総合情報基盤センター・情報基盤システム部門

氏名： 岩間 久和

平成 16 年 10 月 16 日付けで香川大学総合情報基盤センターに助教授として赴任しました岩間久和（いわま ひさかず）です。現在、私は、ゲノム配列情報やマイクロアレイ・データなどの生物学情報をコンピューターを用いて解析する生命情報学と称されるアプローチで研究を進めています。私は、平成元年に岐阜大学医学部を卒業した後、11 年間は臨床医として、臨床研究を行ってきました。臨床研究と生命情報学的アプローチとは、少し距離があるように思われるかもしれませんが。私がこの研究領域に従事するようになったきっかけの一つは、今から 10 年ほど前、実験で必要となったプライマー作成のため、Epstein Barr ウイルスの配列情報を探した時に遡ります。この際に、コンピューター解析によって遺伝子の配列情報から新たな知見を得るという方法論を知り、大変興味をもちました。そして、平成 11 年からは、国立遺伝学研究所で、ヒトの神経伝達物質受容体（ニューロレセプタ）についての遺伝子解析、そして、ヒト・ゲノム情報を用いた遺伝子転写調節領域についての網羅的なコンピューター解析による研究に従事してきました。ヒト・ゲノム解読に代表されるエポックメイキングな時期に、この分野の研究に従事できたことはタイムリーであったと思います。また、日本が主導して進めてきたイネ・ゲノムプロジェクトに参加し、ゲノム情報の解析に取り組むことができたことも幸運でした。

今後、脳神経系や免疫系などの複雑なシステムの特質の解明を目指して、ゲノム情報やマイクロアレイ・データに代表されるような網羅的で大量の生命情報を用い、コンピューターを駆使することにより解析を進め、先進的な知見を得よう研究活動を進めたいと考えています。そのためにも、総合情報基盤センターでは、様々な分野の研究者との協力・協調関係の下、学際的な研究を展開できる基盤の構築を試みたいと思っています。また、このようなアプローチにはコンピューターとネットワーク環境の充実が欠かせませんので、情報基盤の整備に力を注ぎたいと思います。そして、このような活動が教育や地域社会への貢献につながるよう努力していきたいと考えています。



主な論文

- Yokomine T. et al. Structural and functional analysis of a 0.5-Mb chicken region orthologous to the imprinted mammalian *Ascl2/Mash2-Igf2-H19* region. (2005) *Genome Res.*
- Iwama H. & Gojobori T. Highly conserved upstream sequences for transcription factor genes and implications for the regulatory network. (2004) *Proc. Natl. Acad. Sci. USA.*
- Sasaki et al. The genome sequence and structure of rice chromosome 1. (2002) *Nature*
- Iwama H. & Gojobori T. Identification of neurotransmitter receptor genes under significantly relaxed selective constraint by orthologous gene comparisons between humans and rodents. (2002) *Mol Biol. Evol.*
- Iwama H. et al. Epstein-Barr virus detection in kidney biopsy specimens correlates with glomerular mesangial injury. (1998) *Am. J. Kidney Dis.*

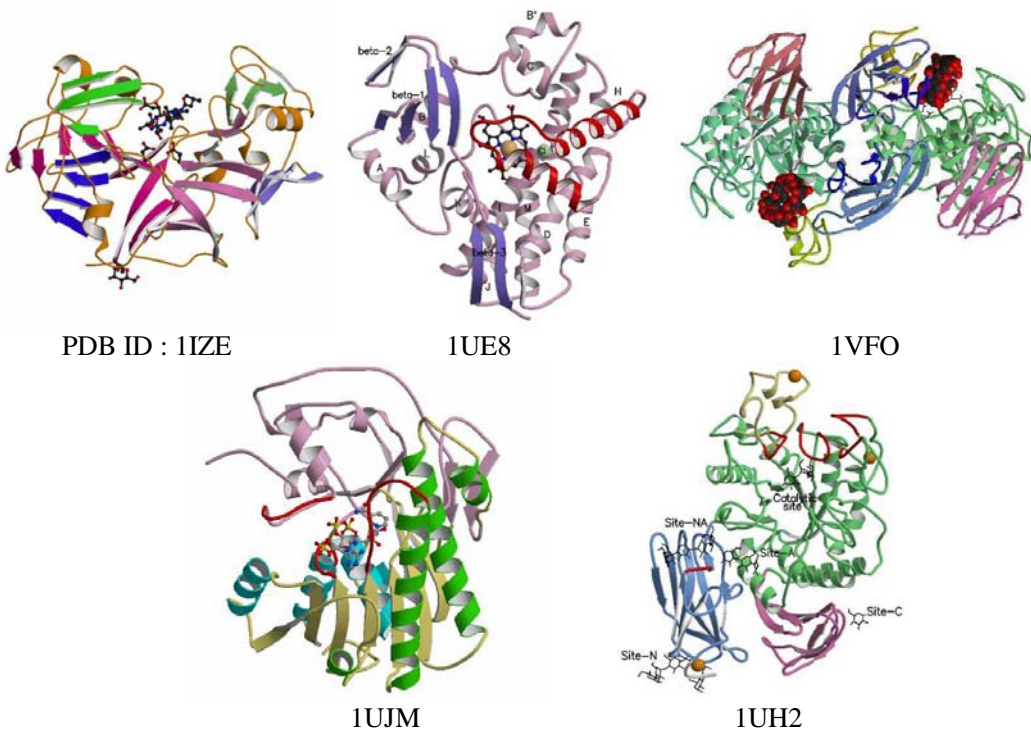
生命現象を分子の3次元構造から解明する

総合情報基盤センター・情報応用システム(分子構造機能学)部門

神鳥成弘

11月1日に総合情報基盤センターに赴任してきました。赴任初年度ということですので、今回は、私の研究分野について簡単にご紹介したいと思います。私の専門は、構造生物学という分野で、生体分子の3次元構造を決定し、そこから得られる3次元構造情報に基づいて、その分子が持つ構造と機能との関係について研究するという学問領域です。手法としてはX線結晶解析を用います。生体を構成する分子は、いくつかの基本構造が組み合わさった複雑なもので、多種多様な機能をもっており、これらの機能は分子の3次元構造と密接に関係しています。分子の3次元構造がわかることにより、それまで「謎」とされていたことが、どんどん解明されてきています。特に、X線結晶解析法は、生化学・分子生物学の発展に大きく貢献し、生化学の教科書を美しく彩っているタンパク質や核酸(DNA, RNA)の3次元構造の大半は、X線結晶解析から得られたものです。

このような時代の流れの中で、私自身、大学院入学から数えると、ちょうど20年間、この分野に身をおき、「3次元構造情報に基づいた分子レベルでの酵素の基質認識・反応機構の解明」というテーマを中心に研究活動を行ってきました。今後は、これまでの研究を、より一層発展させていくとともに、高等動物をターゲットとした新しい研究テーマ、また、実験科学と計算機科学を組み合わせた新たな研究分野にも踏み出していこうと考えています。当研究室に関する詳細な研究内容、研究業績、その他の情報については、<http://www.kms.ac.jp/~xraylab/>をご覧ください。下図は、最近、私たちが構造決定したタンパク質の構造で、すべてタンパク質構造情報データベース Protein Data Bank (PDB)に登録されています。また、こうした専門性を生かして、学部・大学院教育、および総合情報基盤センタースタッフとして学内外の研究者の方々に対する生命情報科学(Bioinformatics)の研究支援にも積極的に携わっていきたいと考えています。今後ともよろしくお願ひします。



縁ありて基盤センターへ

所属：総合情報基盤センター情報応用システム部門
氏名：上原正宏

自己紹介

出自 1945年 昭和20年10月8日(月)生まれ、香川県さぬき市造田出身
学歴 1970年 昭和45年3月 大阪市立大学大学院理学研究科修士課程修了

職歴 1970年 昭和45年4月 大阪府立大学工学部数理工学科
1980年 昭和55年4月 香川医科大学医学部医学科
2003年 平成15年10月 香川大学総合情報基盤センター

専門分野 複素解析学 核函数の理論と応用

抽象的な哲学の一部分、数学の中の複素函数を扱う函数論(複素解析)が専門である。元来、試験官を持つ実験よりも、人文社会的な考えよりも、幾分抽象的な論理的な思考を好み、数学を選んだようである。「人間は考える葦である」と Pascal も云っている。どうい理由か、最近、実学全盛の時代のような。が、コンピュータにも負けない抽象的な論理構築一筋に興味を持ち頑張っている。とは言え、現代社会で、理論しかできないのは大きな欠点となることも承知している。実に残念だが仕方のない事である。

研究概要

解析函数の為すある種の Hilbert 空間には、いわゆる再生性を持つ再生核の存在が知られている。内積を領域積分によるか、境界積分で導入するかにより、その再生核は Bergman 核、または Szego 核とよばれている。また、それぞれの空間には、随伴核である、Schiffer 核、Garabedian 核が存在する。我々は、特に、より広い意味での Szego 核と Garabedian 核の相互関係、それらの新理論の構築、Nehari 問題の解決、より広い現実的な問題への応用等についての研究に興味を抱いている。

これまでに、それらの核についての基本的な特性とある種の応用についての多くの成果を得てきた。今後、これらの核函数の等角写像論や偏微分方程式論の中での具体的な応用についての研究を目指すと共に、これらの解析学を生命情報科学の為の確固たる論理 基盤の構築の一助としたい。

所属学会 日本数学会

2004年9月 日本数学会秋季総合分科会(於 北海道大学)出席

社会貢献 数学雑誌 "Mathematical Review" の Reviewer として、年間1報くらいの割合で Review を出している。

趣味 何時の頃からか、一人落ち込んだときに気分転換を図るもの、ワイワイと仲間を作るもの、この二つのイメージで趣味を開拓してきました。他にも二三ありますが、今現在、実行しているもの、気に入っていること、残っているものは次の通りです。

椿栽培、将棋、落語観賞、古川柳、自転車、灯台巡り

Mail address uehara@med.kagawa-u.ac.jp または math@kms.ac.jp

X線結晶構造学に足をふみいれて

総合情報基盤センター情報応用システム（分子構造機能学）部門

吉田裕美

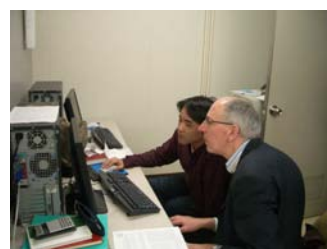
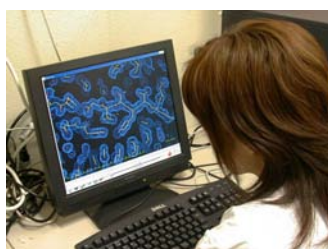
平成16年11月1日より総合情報基盤センター情報応用システム（分子構造機能学）部門助教として着任いたしました吉田裕美です。

これまで私は主な研究テーマとして蛋白質工学を駆使することによってピロロキノリンキノン（PQQ）を補酵素とするグルコース脱水素酵素（Glucose dehydrogenase: GDH）の酵素特性の改良研究を中心にすすめてまいりました。PQQGDHは測定時の酸素濃度に影響されないグルコースセンサー素子、診断用酵素としての応用が考えられていましたが、酵素の安定性、基質特異性の面での酵素の改良が求められていました。この酵素の蛋白質工学的改良を通して、3次元構造が詳細に解明されていない酵素においても、その酵素独自の機能や酵素特有の側面をとらえることによって個別の改良（熱安定性の向上、基質特異性の改良、補酵素との結合の安定性の向上など）を様々な方針で進めていくことができ、蛋白質工学の研究の楽しさから離れることができなくなりました。

近年、個々の酵素特有の性質を理解し蛋白質機能の改良をより効果的にデザインするにはX線結晶構造解析を自分自身が理解すべきであると思い立ち、2年前からX線結晶構造解析を始めました。acyl coenzyme A: isopenicillin N acyltransferase の野生型の構造決定をし、酵素活性を示さない変異型酵素（構造決定済み）との構造比較をすることによって、活性型の野生型酵素と不活性型の変異酵素との大きな構造の違いは何か、そしてその違いが酵素の機能にどのような影響を与えているのか、酵素反応のメカニズムを理解する際にいかに結晶構造解析が多くの情報をもたらしてくれることを実感いたしました。専門は蛋白質工学であり、構造生物学の分野においてはまだまだ未熟なところもありますが、同情報応用システム（分子構造機能学）部門の神鳥教授のもとで様々な蛋白質の結晶構造解析を行い、3次元構造情報にもとづく分子レベルからの機能と構造の相関関係を明らかにしていくことを目的に現在研究を進めています。

また、総合情報基盤センターの教員の方々と協力し、各学部との円滑な連携が図れるよう体制を整え、センター自体が研究を推進していく独自性のある総合情報基盤センターとしての発展を努めてまいりますのでよろしくお願いいたします。

実験風景



生物の進化とゲノムの進化

所属： 総合情報基盤センター

情報教育システム部門

氏名： 竹崎直子

私はもともと人類進化に興味を持ち、生物の進化は基本的には遺伝物質である DNA が変化することであるということが研究活動の根底にありました。しかし、遺伝子あるいはゲノム上の変異というのはもともと1つの個体で起こるものですから、この変異を生物種全体がもつに至る過程については理解できませんでした。その後、集団遺伝学を知ることによってこの謎が解けました。これが現在のゲノム情報科学という私の研究分野へとつながる端緒となりました。これまでは以下に示すような系統解析に関する方法論、主に系統解析による分子進化的解析を行ってきました。今後はゲノムデータを用いた生物のもつシステムの成立とゲノム進化について研究を行っていきたいと考えています。

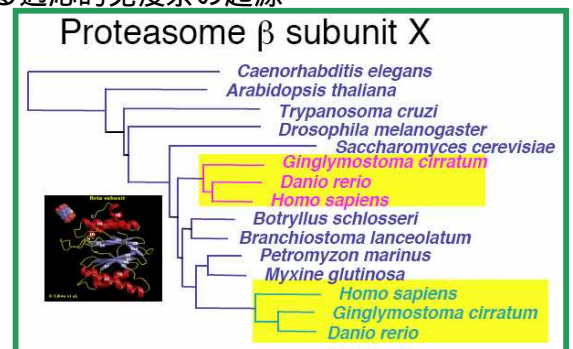
主な研究内容

- ・ 配列データを用いた系統樹作成方法についての理論的研究
- ・ 塩基配列やアミノ酸配列を用いた分岐時間推定方法、置換速度一定性検定方法の開発
- ・ 遺伝子多型データを用いたヒト集団の進化

世界のヒト集団の系統関係

シベリアに固有のヒト集団とアメリカ先住民の祖先（複数回のアメリカ大陸への移住）

- ・ 多くの遺伝子のデータを用いた脊椎動物の系統関係の推定
- ・ 免疫系に關与する遺伝子の系統解析と脊椎動物における適応的免疫系の起源
- ・ アフリカビクトリア湖周辺のカワズメ科の魚の進化
- ・ 脊椎動物における生体システムの起源とゲノム進化



2004年発表論文

Takezaki, N., F. Figueroa, Z. Zaleska-Rutczynska, N. Takahata, and J. Klein. 2004. The phylogenetic relationship of tetrapod, coelacanth, and lungfish revealed by the sequences of 44 nuclear genes. *Mol. Biol. Evol.* 21:1512-1523.

学会発表

“The phylogenetic relationship of tetrapod, coelacanth, and lungfish revealed by the sequences of 44 nuclear genes”, 2004, Genome and Evolution, the joint annual meeting of the International Society for Molecular Biology and Evolution (SMBE) and the American Genetic Association (AGA), State College, Pennsylvania.

キャンパス情報システムにおけるヒューマンネットワークの形成支援

香川大学 総合情報基盤センター
堀 幸雄[†]

1. まえがき

我々はひとの関係を通じて意見を交換し、互いに刺激し合い、新たな発想を生み出す。大学における教育、研究生生活を考えても、誰を恩師とするか、誰と共同研究を行なうかなど、協働関係のありかたは個人の方向性、生産性などを決定づける重要な要因である [7]。さらに人間同士の自然なコミュニケーションを支援する意味でも、ひとを取り巻く人間関係に着目することは重要な方向性である。

キャンパス情報システムにおいても、人間関係を有効に機能させることは学内の知的活動に大きく貢献すると考えることができる。このような仕組みをうまく機能させるためには、キャンパス情報システムにおけるひとの関係を促進する支援技術の役割が重要となる。また情報技術はインターネット環境からいわゆるユビキタスネットワーク環境へと変貌を遂げつつある [1]。このような環境下でユーザの状況に応じた情報支援を行なうには、ユーザの文脈を適切に把握する技術が必要となる。

一方でこの話題に関連する研究も行なわれている。近年 FOAF と呼ばれる人間関係を記述するメタデータの形式が提案されている [5]。FOAF は記述の形式だけであり、こういった人間関係を記述する手間の問題が発生する。またこの情報をシステムとのインタラクションデータから自動的に抽出しようという試みもある。Web のリンク関係からあるトピックに関するコミュニティを抽出したり [3][4]、2 者間の関係を抽出するものである [2]。しかしこういった研究では Web にない情報を扱うことができず、キャンパス内といった閉じたシステムに対して適用することは難しい。

我々はこれまでシステムのアカウンティング情報を用いてユーザの人間関係の抽出する手法について研究を行ってきた [8][9][10][11]。この手法を用いれば、任意の二者間の関係の強さや互いにどういった興味を持っているのかなどを判定することができる。ここではキャンパス情報システムのアカウンティング情報を用いて特定のコミュニティを自動的に抽出する手法について述べる。またキャンパス情報システムでひとの存在、状態を送りあう新しいタイプのコミュニケーションメディアの実現の鍵となる利用者間のコミュニケーション支援について検討する。

[†] horiyuki@cc.kagawa-u.ac.jp

2. データと抽出方法

ここでは学内のキャンパス情報システムの利用者を対象に個々のユーザのシステム内での行動を要素する方法について説明する。

ユーザのシステム内での行動を要素化するには、システムからアカウンティング情報を取得し、各利用者 (x_i) ごとに使用したプロセス (p_i) を重み付けした n 次元のベクトル u_i を用いる (式 (1))。このアカウンティング情報はユーザベースのプロセスの情報を記録・管理している。

$$u_i = \{P(x_i : p_1), P(x_i : p_2), \dots, P(x_i : p_n)\} \quad (1)$$

$$\text{ただし } P(x_i : p_k) = \sum_{j=1}^m w_j^{-1}$$

こうした ID や名前などの単純なデータの他にユーザの利用傾向を示すベクトルをまとめデータベースを作成する。このデータベースには、ユーザから属性値へのマッピング $f_a : Q \rightarrow X_a$ や属性値からユーザへの逆マッピング $f_{key}^{-1} : X_{key} \rightarrow Q$ もハッシュを用いて高速に検索できるようにする。これにより、ユーザ A のある興味、属性を引く、つまり $f_{興味}(f_{名前}^{-1}(A))$ を求めるといったことが簡単かつ高速に行なうことができる。

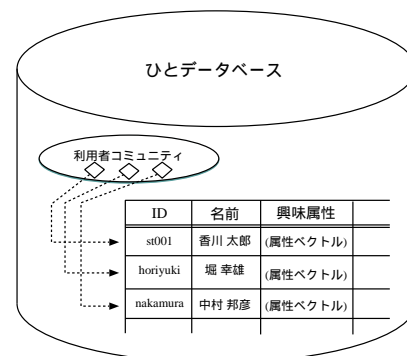


図 1: ひとデータベース

2.1 人間関係の抽出

キャンパス情報システムは研究、課題、演習などを行なうことを目的とし、さらに学外ネットワークから利用できない閉じた環境にあるため、物理的に計算機室において利用することになる。計算機室を利用する人が多いほど、計算機室が賑やかになるのはコミュニケーションを行なう友人が近くに居ると

考えられる(図2)．したがって隣席状態をなんらかの関係のある利用者同士と仮定して、人間関係ネットワーク構造を抽出する．



図2: うるさい隣席

このときのユーザ間の共起の強さを Simpson 係数により求める(式(2))．このユーザをノード、隣席関係をエッジとして得られたグラフを利用者間のコミュニティグラフとして扱う．

$$R(X, Y) = \begin{cases} \frac{|X \cap Y|}{\min(|X|, |Y|)} & \text{if } |X|, |Y| > k \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

3. 関係構造の評価と分析

これまで本手法で抽出した関係構造についての分析を行なった．抽出したグラフを図3に示す．本学においてはユーザ同士のつながりを示すシステムを開発段階である．以降に示すのはこれまで得られた評価値である．

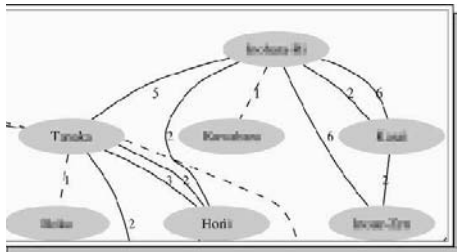


図3: 抽出した関係ネットワーク

また抽出した関係の精度を確かめるためにアンケート回答による評価を行なった．表1にアンケート回答による適合率、再現率を示す．

表1: アンケートによる評価

	適合率	再現率
抽出した関係	93.5%	82.0%

適合率に比べ、再現率は低いのはすべての情報がアカウント情報にあるわけではないためと思われる．しかし9割程度の再現率で関係を抽出できると点は、キャンパス情報システムにおけるコミュニケーション支援という目的には役に立つだろう．

3.1 関係構造の分析

次の得られた関係グラフに対して、構造的特性の分析を行なった．関係のパターンが規定する各種指標はグラフに描かれる関係を行列に変換して計算され、定量的に算出される．近接性(N)、結合関係(D)、構造同値性(S)などが代表的なものだが、それぞれからコミュニティの

分類、中心人物、またグループ内およびグループ間の差などを特定することができる．

$$N_{ij} = \text{shortest_path}(i, j) \quad (3)$$

$$D_{ij} = \frac{\sum_{i=1}^N \sum_{j=1}^N (x_{ij} + x_{ji})}{N^2/2} \quad (4)$$

$$S_{ij} = \sqrt{\sum_{k=1}^N [(x_{ik} - x_{jk})^2 + (x_{ki} - x_{kj})^2]} \quad (5)$$

コミュニティの関係構造を分析すること関係的な要因がわかり、行動の予測やコミュニティの組織化プロセスの理解やその動態をエージェント・シミュレーション[6]などを行なうことが可能である．

4. まとめ

本研究は人間関係ネットワークをキャンパス情報システムから抽出する手法を提案したが、近年のユビキタスコンピューティング環境の観点からユーザの位置、活動情報、欲求、嗜好を考慮しユーザに応じた情報支援といった研究に近い．

流行や情報、知識の流れという点において、コミュニティがどう存在するのかを特徴づける方法や、本研究で得た利用者を取り巻く人間関係から、利用者の活動情報、欲求、嗜好、これまでの履歴を考慮し、利用者の文脈に応じた情報支援の方法など今後様々な研究が可能であると考えられる．

ただし、いくつか考慮しなければならない点がある．アカウント情報に関する問題¹、利用者のプライバシーに関する問題など、社会的な合意がまだ不十分でむづかしい問題である．これらの情報を元にサービスを提供する際には十分な配慮を行なうことが必要である．

本研究はキャンパス情報システムにおける活動を分析、促進するといった点でも重要な基礎データになる．また最近のソーシャルネットワークサービスに関わる話題でひとのネットワークが取り上げられることが多い．本研究で提案した抽出手法は、こういった話題へのひとつの有効なアプローチであると考えている．

参考文献

- [1] 中島秀之, 車谷浩一, 伊藤日出男: ユビキタス情報処理による社会支援, 情報処理, Vol. 45, No.9, pp.907-911, 2004.
- [2] 松尾 豊, 友部 博教, 橋田 浩一, 石塚 満: Web からの人間関係ネットワークの抽出と情報支援, 人工知能学会全国大会, 1F1-02, 2003.
- [3] Kleinberg, J.M.: Authoritative Sources in a Hyperlinked Environment, *Proc. ACM-SIAM*, pp. 668-677, 1998.
- [4] Kumar, S. R., Raghavan, P., Rajagopalan, S., and Tokins, A.: Trawling the web for emerging cyber communities, *Proc. 8th WWW Conference*, 1999.
- [5] Dan Brickley, Libby Miller: FOAF Vocabulary Specification, <http://xmlns.com/foaf/0.1/>
- [6] Kurumatani, K.: Mass User Support by Social Coordination among Citizens in a Real Environment., *Multi-Agent Mass User Support, Lecture Notes in Artificial Intelligence 3012*, Springer, pp.1-17, 2004.
- [7] 安田 雪: ネットワーク分析, 新曜社, 1997.
- [8] 石川 大介, 堀 幸雄: 端末利用者間におけるグラフの分析, 情報処理学会, 第64回全国大会, 2002.
- [9] 堀 幸雄, 石川 大介: 端末利用者間のコミュニティ支援, 情報化学技術フォーラム 2003, 情報処理学会.
- [10] 田代 淳一, 堀 幸雄, 後藤 英一: アカウンティング情報を用いた利用者の専門性分析, 情報処理学会, SIG-IS, No.83, Vol.2003, 2003.
- [11] 堀 幸雄, 米倉 正和, 後藤 英一: 情報システム利用者における small-world 構造の検証, 情報知識学会, Vol.14, No.2, pp.49-52, 2004.

¹ アカウンティング情報はもともと課金目的のために設計されたものである

第1回総合情報基盤センター先端研究セミナー開催報告

総合情報基盤センター情報応用システム(分子構造機能学)部門

吉田裕美・神鳥成弘

2004年11月22日にオランダのフローニンゲン大学からダイクストラ教授を招いて第1回総合情報基盤センター先端研究セミナーを行いました。

「The breaking and making of glycosidic bonds by amylases and chitinases (アミラーゼとキチナーゼによるグリコシド結合の分解と形成)」という表題のもとに、機能性環状オリゴ糖であるシクロデキストリンの合成・分解に関わる酵素であるシクロデキストリングリコシルトランスフェラーゼ (Cyclodextrin glycosyltransferase; CGTase), およびキチンの (1,4) 結合を切断する酵素であるキチナーゼの2つの酵素について講演をしていただきました。講演内容は、ダイクストラ教授のグループにより構造決定の成果を中心に、そこから得られる3次元構造情報に基づいた酵素反応機構の解明、活性触媒部位の電子状態にまで及び、大変興味深いものでした。また、最後に、Gauchers 病の診断マーカーとして認識されているヒト由来の chitotriosidase の基質結合部位の構造情報についても話していただき、タンパク質X線結晶解析という基礎的な研究分野が、どのように臨床医学に貢献できるか、その可能性についてまで言及していただき、社会的にも興味深い講演会となりました。

当日の参加者は多くはなかったのですが、聴衆者からの質疑も活発で講演後は熱心なディスカッションが行われました。

今後もこのような先端研究セミナーを続けてまいりますので皆様どうぞふるってご参加ください。

Bauke W. Dijkstra 教授のプロフィール

氏名: Bauke W. Dijkstra (バウケ・W・ダイクストラ)

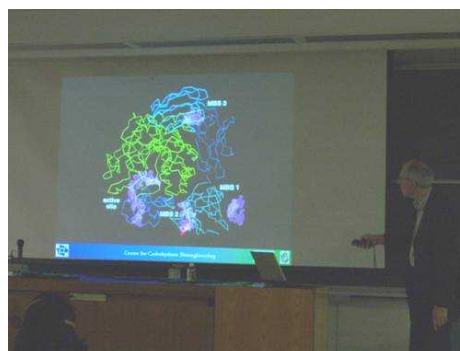
現職: full professor of Biophysical Chemistry (生物物理化学科 教授)

所属: Laboratory of Biophysical Chemistry, University of Groningen,
The Netherlands (オランダ, フローニンゲン大学 生物物理化学研究室)

関心のある研究分野:

蛋白質の結晶構造解析, 酵素反応機構, 蛋白質工学, 宿主・病原体相互作用

ホームページ: <http://www.xray.chem.rug.nl/>



教 育

分子生物学(医学部医学科2年生後期)講義の概要の紹介

総合情報基盤センター

竹崎直子・岩間久和・神鳥成弘・吉田裕美

情報教育システム部門 竹崎直子
情報基盤システム部門 岩間久和

現在バクテリアからヒトまで様々な生物のゲノムが解読されているが、生物が生体を形成する上での基本的な情報はゲノム上に記載された情報であり、ゲノム情報が変化することによって生物は変化していく。従って、生物を理解するうえでゲノム情報を理解することが今後ヒトを含めた生物のもつシステムを理解していく上で必要であり、またゲノム情報は今日の医療にも密接に関わっている。

講義ではゲノム上の遺伝情報とはどういうものであるのか、生物種によってどのような違いがあるのかなどゲノム構造についての紹介、異なる生物のゲノム情報の違い、ヒトなど同生物種内での個体差の基礎となるゲノム情報の個体間の変異がどのようにして蓄積するのかについての序論的解説、ゲノム情報の個体間の変異を用いた疾患遺伝子の探索方法についての解説を行った。

- 1-2回 遺伝子とゲノム
- 3-4回 遺伝学
- 5-6回 遺伝子の変化と進化
- 7-8回 疾患遺伝子

情報応用システム部門

神鳥成弘・吉田裕美

バイオテクノロジー技術の目覚ましい発展にともない、分子生物学の進歩は著しく、新しい知見も増え続けている。しかし、分子生物学において根本的に理解しなければならない概念や基本的な知識を身につけていけば、新しい情報を得たときに対応でき、また最先端の研究を理解することができる。

分子生物学は生化学と非常に関連の深い領域であることから、学生諸君にとって馴染みやすい内容であると思われるので、ここではわれわれの専門分野である構造生物学の立場からみた分子生物学の基礎を中心に理解を深めてもらうことを目的に12月から2月にかけて講義を行った。教科書としてはヴォート「生化学」を中心に、蛋白質の構造情報をまじえながらスライドを用いて視覚的な理解をできるようにした。

今年度、当部門が担当したところは以下の通り。

- 1-2回 核酸の化学構造(構造と操作)
- 3-4回 核酸の配列決定(PCRとDNAシーケンスの原理)
- 5-6回 転写
- 7回 翻訳
- 8-9回 DNAの複製と修復

教育活動

所属：総合情報基盤センター情報応用システム部門
氏名：上原正宏

教育指針

学生が学問に立ち向かう様は、あたかも、古代原始人があの巨大なマンモスに挑む姿に似ている。その時、原始人達は一体何を為したであろうか。ある者は石斧を作り、矢尻を研いたことだろう。またある者は、古老に相談をし、叡知を磨き、策略をこうじたであろう。自然科学の言語、文法とでも云うべき数学には、その抽象的な論理構成の困難さが常につきまとう。学部教育に当たる時、受講する学生の目指す専門が医学であっても、農学であっても、人文社会であっても、いわんや数学であっても、何であっても、一般的には、数学の抽象的で難解とされる論理の必然性と論理展開をできる限り優しく、丁寧に説き、その具体的な応用例を数多く示すことにより、数学という学問の美しさ厳しさを紹介することは勿論のこと、多くの時間を費やしてでも、その楽しさ、その有用性を理解させると共に、人間としての叡知を、考える勇気を身に付ける重要性を認識できることを最大の目標としている。

平成 16 年度授業担当科目とその内容

前 期

(1) 応用数学 (医学部 2 年生) 選択

内容 微積分の特に極小値、極大値問題への具体的な応用、基本的な微分方程式の紹介とその解法、医学における数学的な解析的な思考法の紹介、基礎、臨床における微分方程式の取り組みの紹介、具体的な現実の問題に対し微分方程式を応用する。

目的 抽象的で、難解とされる数学の分野の中の、微積分、微分方程式という解析学の基本的な分野においても、現実の問題に対して様々なアプローチがあり、多くの応用があることを理解させる。

(2) 統計学 (医学部 2 年生) 必修

内容 統計学の基本的な事項、確率変数、確率分布の導入、二項分布、正規分布の基本的な性質、不変推定量の必然性、区間推定、検定の三兄弟の紹介、分割表による検定の必要性、医学における初歩的な検定の応用を計る。

目的 微積分の原理が理解できることを仮定して、統計学の初歩的な仕組みを理解させ、その統計学の理論が、具体的な現実的な問題に応用できることを確かめる。基本的な区間推定とパラメトリック検定が実行できる。

(3) 数学 E (全学教養) 共通

内容 行列と行列式の基本的な性質、行列と行列式の新しい概念に慣れる。またそれらの違いが具体的に理解できる。余因子と余因子展開、Cramer の公式、掃出し法などにより、連立一次方程式の解法と解の分類ができる。

目的 高校までの知識から、より高度な数学の帰納的発見の経過を紹介する。その過程を詳細に見ることから、より深い理論の構築ができることを、行列、行列式を中心に確かめる。新しい基本概念の導入の必然性を理解する。全ての連立一次方程式の解の分類ができる。

(4) 現代解析学 (教育学部4年生)選択

内容 初等的な解析学を履修した者に対して、学部における解析学の根幹の思想の紹介。極限の連続性について反例を多く示す。一様収束の概念の必要性、一様世界の必然性、項別微分、項別積分の可能性等について言及する。

目的 何気なく計算していた極限、微積分の中で、気付かなかった反例を紹介し、解析学における論理的に疑問を抱かせる。その疑問を解消するためのさらなる論理展開の必要性を数学的な歴史と共に紹介し、優れた先人の足跡を追う。解析学の理論の必然性を説き、数学に対するさらなる興味を抱かせる。

後 期

(5) 生命科学D (全学教養)共通

内容 簡単な線形常微分方程式を導入し、それらの具体的かつ簡便な解法の解説する。解析学の医学への具体的な応用、および医学での微分方程式の例題を数多く紹介。Laplace 変換の性質と初期値の与えられた定数係数線形微分方程式の解法への応用を計る。

目的 現実の多くの現象を微分方程式でとらえることが可能であることを認識させる。さらに、高校程度の微積分の知識だけからでも、多くの医学への応用があることを紹介し、さらなる数学への興味を深く抱かせる。

(6) 函数論 (教育学部3年生)選択

内容 複素数と複素平面の紹介。函数論に必要な二変数函数の微積分の復習。初等的な複素数函数の導入とその特性の解説。正則函数の基本的性質、Cauchy の積分定理、積分公式、留数原理の意味、複素解析の実定積分への具体的な応用の解説をする。

目的 実数の世界とは異なる複素数の世界の美しさを紹介することは勿論のこととして、複素函数の理論をかいま見ること、解析学の新しい事実を知り、複素数を經由することで、実数の世界がより明白になることを認識させる。

集中講義「知的 CAI」

所属： 総合情報基盤センター

氏名： 林敏浩

著者の研究専門領域は教育工学であり、これに関連する講義を学部・大学院に提供している。今年度は、平成 16 年 12 月に徳島大学大学院工学研究科知能情報工学専攻の博士前期課程学生に対し集中講義「知的 CAI」を実施した。本稿では、当該講義に関して報告する。

知的 CAI は「教授と学習の教育現象を人工知能技術により明らかにし、高度個別教育環境を実現する CAI」であり、教育・訓練のための CAI、認知科学、さらに計算機科学（特に人工知能）を含む複合領域を形成する。「知的 CAI」の集中講義は以下のような内容で実施した。

1. CAI 概説：CAI の定義・目的・歴史・用途・アーキテクチャ
2. CAI 詳説：各種伝統 CAI システムのアーキテクチャと特徴
3. 知的 CAI 概説：知的 CAI の定義・歴史・アーキテクチャ・基本モジュール
4. 知的 CAI 詳説：代表的な知的 CAI システムの紹介・近年の知的教育システム

また、本講義では、講義の様子をビデオ撮影し、集中講義資料（PowerPoint 資料）と連動する e-Learning 教材を徳島大学側で作成した。本教材はインターネット上で限定公開され、講義に出席できなかった学生がこれを用いて自主学習を行ったり、課題レポートを作成する学生が復習用教材として利用した。図 1 は作成した e-Learning 教材のスナップショットである。

なお、希望いただければ著者は本学の学部・大学院生向けに教育工学関連の講義を提供したいと考えている。興味のある方は著者（hayashi@eng.kagawa-u.ac.jp）まで連絡されたい。



図 1：集中講義「知的 CAI」の e-Learning 教材

高校生へのセンター紹介

所属： 総合情報基盤センター

氏名： 竹崎直子・岩間久和

平成16年11月4日、医学部キャンパス訪問の丸亀高校1年生17人に対して以下のよう
に総合情報基盤センターの施設見学および概要の説明、研究内容の紹介を行った。

午後2時～2時30分 総合情報基盤センター施設見学

河内、川田、池田のセンター職員の協力のもとに、センター内のウェブサーバー、
メールサーバーなどのサーバー類、ネットワークケーブル接続ノードなど医学部キャン
パスにおける情報基盤の紹介を行い、電子メール、インターネットなど香川大学での各学生
が受ける情報基盤サービスなどについても簡単にふれた。

午後2時～2時30分 生命情報科学の紹介

「遺伝子とゲノム」というタイトルで生命情報科学の基礎となる遺伝子、ゲノムの
構造、遺伝情報の伝達についての簡単な紹介を行った。



センター施設見学



「遺伝子とゲノム」講義風景

遺伝子とは？

遺伝情報を親から子に伝える遺伝物質の単位

ゲノムとは？

ある生物がもつ一組の染色体上の全ての遺伝物質

遺伝子 gene
ゲノム genome

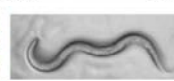
Eucarya (真核生物)	ゲノムサイズ(bp)	遺伝子数
菌類 <i>Saccharomyces cerevisiae</i> (酵母)	12 million	6300
植物 <i>Arabidopsis thaliana</i> (シロイヌナズナ)	125 million	26000
動物 <i>Caenorhabditis elegans</i> (線虫)	97 million	19000
<i>Homo sapiens</i> (ヒト)	2800 million	30000
<i>Mus musculus</i> (マウス)	2800 million	30000
<i>Takifugu rubripes</i> (フグ)	365 million	30000



酵母



シロイヌナズナ



線虫



マウス



フグ

キャンパスウォークにおけるセンターの紹介

香川大学 総合情報基盤センター
堀 幸雄†

1. キャンパスウォークについて

香川大学は2004年度より下記の日程でキャンパスウォークを実施することになりました[1]。キャンパスウォークとは受験生、一般の方を対象に香川大学を広く理解して頂くとともに、特に受験生には本学に対して興味を持ってもらうためのものです。県内をはじめ、岡山、徳島、高知、大阪など県外の高校生の総勢91名に参加者していただきました。

表 1: キャンパスウォーク開催日程

日程, 開催時間 (13:30 ~ 16:00)	イベント
平成 16 年 7 月 21 日 (水)	大学紹介, 学科専攻紹介, 施設見学, 入試課担当者による入試個別相談, 進路相談会, 在学生への学生生活等の相談, 体験学習, 研究室紹介など
平成 16 年 8 月 4 日 (水)	
平成 16 年 8 月 18 日 (水)	
平成 16 年 8 月 25 日 (水)	
平成 16 年 9 月 1 日 (水)	
平成 16 年 9 月 15 日 (水)	
平成 16 年 10 月 6 日 (水)	
平成 16 年 10 月 20 日 (水)	

2. 本センターの紹介

本センターの紹介として施設紹介, コンピュータルームの見学, そしてセンターの目的, 業務内容, 私が行っている研究についてわかりやすく解説, 説明をしました。下記にそのときの様子を紹介します。



図 1: 総合情報基盤センター キャンパスウォーク時の様子

参考文献

- [1] キャンパス・ウォーク (大学見学会) の実施, 香川大学ホットニュース, 8/4, 2004.
<http://www.kagawa-u.ac.jp/univ/koho/20gou/campus%20walk1.htm>

† horiyuki@cc.kagawa-u.ac.jp

情報基盤

香川大学のネットワーク構成

所属： 総合情報基盤センター

氏名： 林敏浩、河内一芳、瀬野芳孝、曾根計俊

平成 15 年 10 月に旧香川大学と旧香川医科大学が統合されたことにより、現在の香川大学は

幸町地区、三木医学部地区、三木農学部地区、林町地区の 4 つのキャンパスから構成される。これらキャンパス間は図 1 のように ATM ネットワークで相互接続されている。香川大学のキャンパスネットワークの基幹部は総合情報基盤センターが管理している。各地区ではそれぞれネットワークを管理する組織（例えば、工学部では情報基盤室）が形成されて、総合情報基盤センターと連携し、キャンパスネットワークが分散運用されている。

各地区は、ATM メガリンクに対し、ATM スイッチまたは ATM ルータで接続され、そこから教育系 LAN や事務系 LAN が接続されている。三木町医学部地区はさらに医学部 LAN からファイアウォールを介し病院 LAN が接続されている。幸町地区にある総合情報基盤センター幸町分室から、附属小学校・中学校、附属農場などがネットワーク接続されている。また、同分室よりファイアウォールを介し SINET に接続されている。

なお、現在、策定が進んでいる本学情報セキュリティポリシーを考慮して、本稿ではネットワークの説明は概説にとどめる。図 1 も概要である点に注意されたい。このため、詳細なネットワーク構成に関して知りたい場合は総合情報基盤センターに問い合わせをされたい。

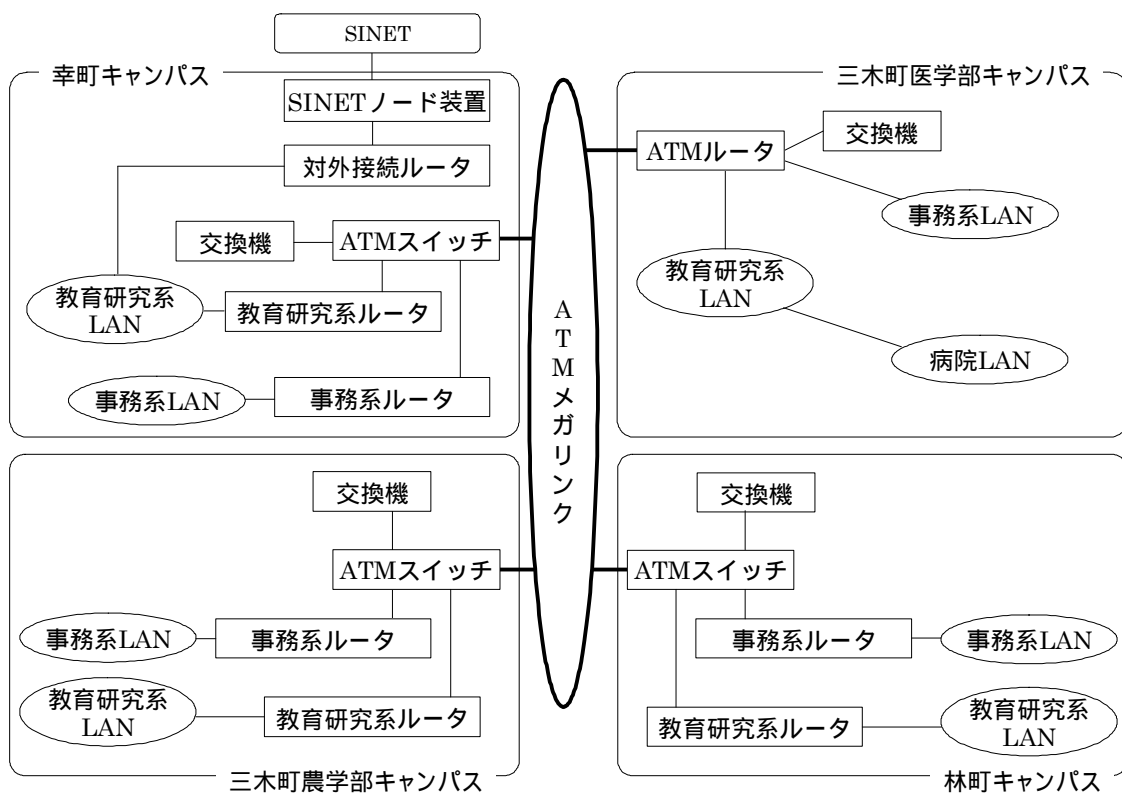


図 1：香川大学のネットワーク構成

総合情報基盤センター業務・サービス

所属：総合情報基盤センター（医学部キャンパス）

氏名：河内 一芳・川田 延枝・池田 裕子

総合情報基盤センターは、平成15年10月に旧香川大学と旧香川医科大学の統合時に設立された新しい組織で、医学部キャンパスの基礎臨床研究棟2階にあり、また同キャンパスでは乗松センター長をはじめ専任教員5名・技術専門職員1名・他非常勤職員2名が業務にあたっています。基盤センターの業務は細かく挙げればきりが無い程さまざまありますが、第一は医学部キャンパスのネットワークの安定稼働ではないでしょうか。

そのため下記のようなことをしています。

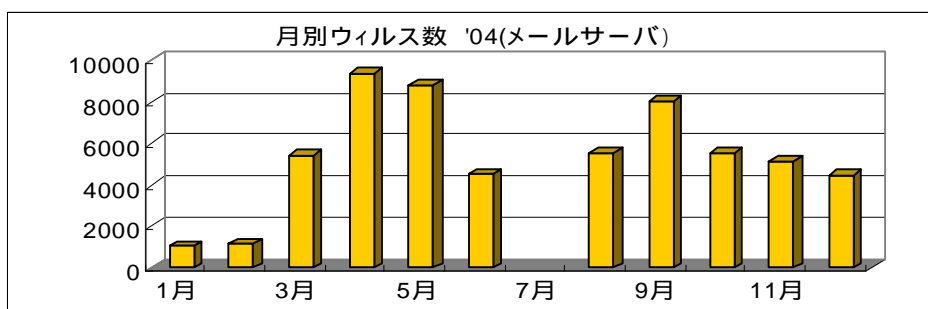
- 1) インターネットへの接続口には Firewall を設置してアクセスの制限を行い、さらに附属病院の医療情報ネットワークと医学部ネットワークの間にも Firewall を設置して厳しく制限しています。
- 2) 侵入検知機器による不正アクセスの防止や、ネットワーク監視ソフトによる各 Switch 類の状況監視を常時行い、障害が発生した場合はすぐに対応出来るようにしています。
- 3) MRTG(Multi Router Traffic Grapher)というソフトを使って Firewall やルータを流れるトラフィック量の監視を行い、異常があればただちに原因を究明し対策を講じています。またこのデータを今後のネットワーク機器の更新時に役立てることにしています。
- 4) Web サーバや Mail サーバなど各サーバ類にはサーバ用ウィルス対策ソフトをインストールし、毎日のように新種が発見されるコンピュータウィルスに対応しています。
(資料1参照)
- 5) その他ネットワークの高速化を図るためそれに対応した機器の導入や、計画立てて機器の更新を図りネットワークの整備を進めています。

サーバダウンやウィルス感染などネットワークに障害が発生した場合は、各学部や基盤センター以外の教員とも連携して、迅速に対応出来るようにしています。

医学部キャンパスでは、学生用無線 LAN アクセスポイントを講義実習棟や図書館など19カ所に設置して学生のインターネット環境の向上を図っています。また以下の医学部ネットワーク管理室の業務もかねています。

- 1) 学生を含めたネットワーク利用者からの各種申請書（電子メール利用申請書・情報ネットワーク接続申請書・無線 LAN 利用申請書等）にもとづき、メールアカウントの発行や IP アドレスの付与等の処理（資料2参照）
- 2) ネットワークに関する各種アプリケーションソフトのインストール・バージョンアップのサポートや LAN ケーブルの作成、情報のアナウンス等のサービス
- 3) 年1回アンケートを実施しネットワーク接続機器の状況を把握し IP アドレスの管理やメールアカウントの管理

(資料1)



このグラフはメールサーバにインストールされているウイルス対策ソフトにより検出・削除された月別ウイルス数です。(6・7・8月は実数不明)

(資料2) ネットワーク利用状況(医学部・平成16年12月現在)

・ 接続端末数	教育・研究用	約1600台
	医療情報用	約1200台
・ メールアカウント発行数	教職員(医師を含む)	950
	研修医・研究生	100
	学生(院生を含む)	1000
・ 無線LAN利用申請者(学生)		70名

幸町分室の紹介

総合情報基盤センター

堀 幸雄，瀬野芳孝，曾根計俊，高橋多恵

建物

総合情報基盤センター幸町分室は幸町キャンパスの中、経済学部構内の少し奥まったところにあります。

建物は茶色っぽい色の2階建てで、煉瓦タイル模様の外壁が目印です。昭和49年(1974年)に計算センターとして建てられ、平成3年(1991年)の情報処理センター発足に合わせて一部改修して、現在も引き続き総合情報基盤センター幸町分室として使っています。

経済学部3号館すぐ横の北側にある入り口が玄関です。玄関を入れてまっすぐ進むと突き当たりが事務室になります。

業務・サービス

幸町分室では技術的な業務のうち、主に幸町地区に関係するところを担当しています。

“技術的な業務のうち幸町地区に関する部分”といってもピンとこないかもしれませんので、いくつか代表的なものを紹介します。

授業、レポート作成、履修登録など直接的な授業支援、そして、インターネットを検索し種々の情報を入手したり、就職活動に利用したりといった学生生活支援のため、幸町地区には合わせて237台のパソコンを設置しています。これらのパソコンは授業での使用を第一の目的としていますので、授業に支障を来さないよう常に最良の状態に維持管理する必要があります。この維持管理には、機器の機械的な故障はもとより、OSやアプリケーションソフトウェアのセキュリティ対応も含まれます。また、授業で使うソフトウェアの選定と導入(インストール)も欠かせない業務です。さらに、ネットワーク上のセキュリティを確保するため利用者資格があるかどうかの確認も重要となっており、新入生の入学や卒業といった学生の異動時の利用者登録作業も大きな業務の一つです。

パソコン室は授業での利用を優先しますが、授業のない時間帯は自習用に解放しています。自分のあいている時間帯が授業と重なった場合には時間にせかされることなく使えるよう、24時間利用可能なサービスを提供しています。幸町キャンパスでは全学の教養教育を担当していて、全学生の約6割程度が学んでおり、パソコン室利用時の混雑緩和にも効果を発揮しています。休日・夜間の利用ではセキュリティの確保のため、入室の際学生証による認証が必要です。その認証システムの運用管理も幸町分室が行っています。

一方、研究や日常業務でパソコンを使うことが必要不可欠になっていますが、ネットワークの介在なくしてはただの箱にもなりかねません。そのネットワークの正常運用も非常に重

活動報告

アタックレポートの集計

教育学部 青木昌三
総合情報基盤センター幸町分室
瀬野芳孝 曽根計俊 高橋多恵

平成15年8月にWindowsのセキュリティホールを攻撃して、感染させるワームが大流行した。感染したパソコンはランダムなアドレスにパケットを送信して、セキュリティホールのあるパソコンを感染させる。感染したパソコンは同様のことを順次繰り返す。その結果、ネットワーク全体に膨大な数のパケットが飛び交い、ネットワークの処理能力を超え使用できなくなる。

本学においてもネットワークは2日間使用できなくなった。対策についていろいろ検討したが、現在のネットワーク機器では対応できず、結論として各部局のネットワーク担当者に、当該部局のワーム感染パソコン1台1台への対処方を担ってもらうことになった。当初は感染したパソコンを特定するために、常時稼働させているモニター用パソコンのパーソナルファイアウォールのログを解析し、このパソコンを攻撃してくるワームに感染したパソコンのIPアドレスを割り出しては、各部局のネットワーク担当者に定時にメールを送信していた。だが、こうしたログ解析によるワーム感染パソコンを特定する方法ではあまりにも効率が悪いので、LINUXサーバにおいてこうしたログ解析処理等の自動化を行った。感染したパソコンからLINUXサーバに攻撃してくるパケットをログに記録して、定時に各部局のネットワーク担当者にメールを自動送信するのである。

以下はネットワーク担当者へのメールのサンプルと2004年のLINUXサーバへの攻撃パケット数をグラフ化したものである。現在でも感染したと思われるパソコンからの攻撃が多数見受けられる。

ネットワーク担当者へのメールのサンプル

攻撃があれば、その時から定時(9時、12時、15時、18時、21時、24時)に1日間送信される。

関係者各位

総合情報基盤センター幸町分室です。

センターのメールサーバに対して攻撃を行うPC等のログをお送りします。対応の程、宜しくお願いします。

***** 攻撃レポート -->135/TCP,445/TCP *****

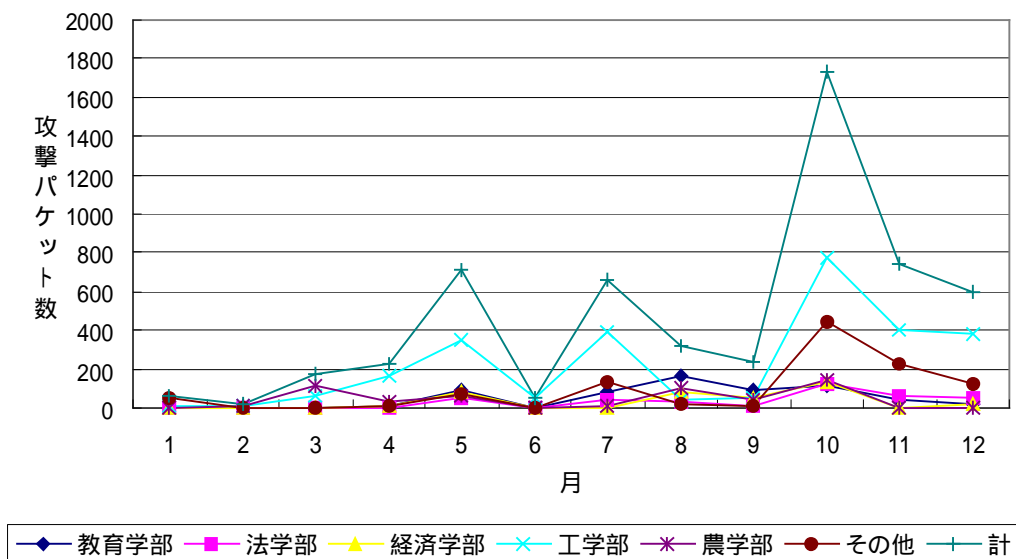
日時,送信元 IP アドレス,送信先 IP アドレス,支線管理部局:支線場所

Jan 10 17:31:26,133.92.xx.xxx:2884,133.92.xx.xxx:445,XX 学部:XX 研究棟
Jan 10 17:31:29,133.92.xx.xxx:2884,133.92.xx.xxx:445,XX 学部:XX 研究棟
Jan 10 17:48:24,133.92.xx.xxx:3142,133.92.xx.xxx:445,XX 学部:XX 研究棟
Jan 10 17:48:27,133.92.xx.xxx:3142,133.92.xx.xxx:135,XX 学部:XX 研究棟
Jan 10 18:25:46,133.92.xx.xxx:2907,133.92.xx.xxx:135,XX 学部:XX 研究棟

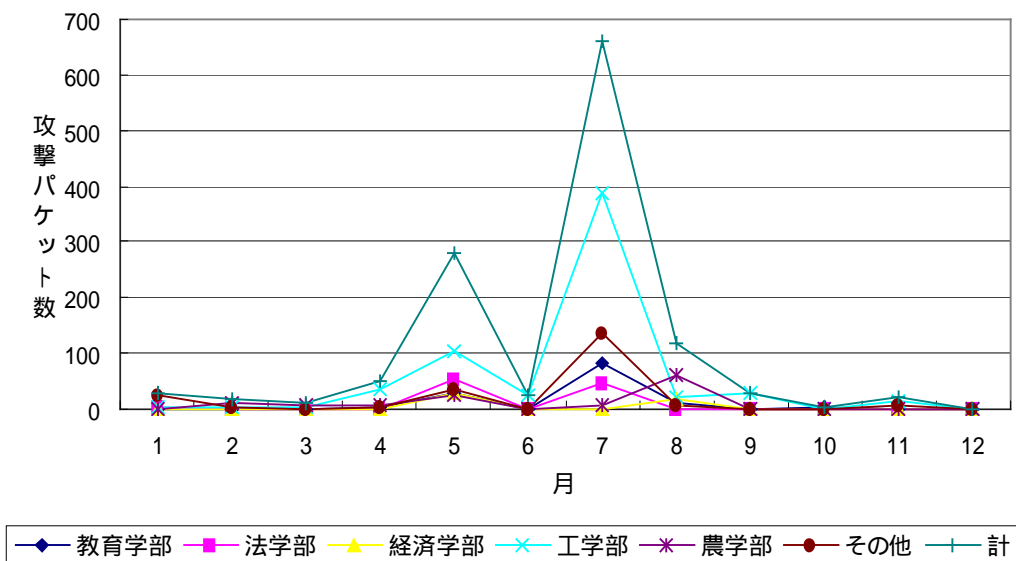
2004年のLINUXサーバへの攻撃パケット数

今回調査対象としたのは2種類のパケットで、ポート135/TCPと445/TCPを攻撃するものである。以下、全体、ポート別にLINUXサーバに攻撃を加えてきたパソコンのパケット数を部局毎に集計したグラフを示す。

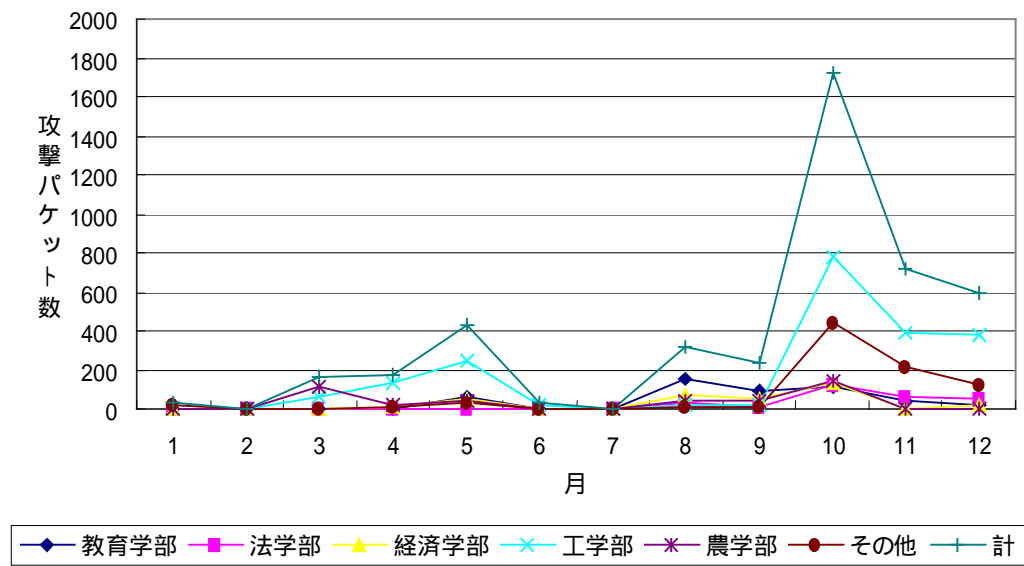
攻撃パケット数全体推移(2004/1~12)



攻撃パケット数ポート135/TCP推移(2004/1~12)



攻撃パケット数ポート 445/TCP 推移(2004/1~12)



参考

Windows 特有のサービス

135/TCP RPC (Remote Procedure Call)

137/TCP NetBios の名前解決

138/TCP NetBios のブラウジング

139/TCP NetBios セッションサービス

445/TCP ダイレクトホスティング SMB

ウィルスチェックサーバによるアラートメールの集計

教育学部 青木昌三
 総合情報基盤センター幸町分室
 瀬野芳孝 曾根計俊 高橋多恵

昨年度に引き続き、総合情報基盤センター幸町分室に導入されているウィルスチェックサーバにより、2004年1月から同12月までの間にメールから除去されたウィルス件数やウィルスの種類について集計した結果を報告する。

(a)ウィルス除去数ベスト10(2004/1~12月)

2004年1月から12月までの間でのウィルスチェックサーバにより除去されたウィルス数について、12ヶ月間の合計数が多い順に上位10のウィルスと「その他」(後出の(b)の項参照)をまとめたものをリスト1に、関連したグラフを図1~図4に示す。

リスト1：ウィルス除去数ベスト10

	1	2	3	4	5	6	7	8	9	10	11	12	計
WORM_NETSKY.P	0	0	2493	12339	6930	7182	7186	8280	6706	8357	7245	8276	74994
HTML_Netsky.P	0	0	0	4626	6761	4899	4786	5558	4224	4863	4416	5126	45259
WORM_NETSKY.Q	0	0	3697	13760	7662	3541	3465	3602	2212	2042	2202	2418	44601
WORM_NETSKY.D	0	0	8114	4532	3488	3249	3330	3873	2728	3196	2173	2505	37188
WORM_MYDOOM.A	1307	12704	93	1	1	0	3	2	97	41	0	52	14301
WORM_LOVGATE.W	0	0	0	1729	694	1180	636	644	707	613	304	1845	8352
WORM_MIMAIL.R	4690	0	0	0	0	0	0	0	0	0	0	0	4690
WORM_NETSKY.C	0	45	2403	571	305	261	168	138	97	124	276	159	4547
WORM_NETSKY.Z	0	0	0	82	539	427	590	547	696	167	124	230	3402
WORM_RATOS.A	0	0	0	0	0	0	0	3239	0	0	0	0	3239
その他	652	521	1175	1078	1781	2271	1693	1461	1391	2192	3081	1293	18589
合計	6649	13270	17975	38718	28161	23010	21857	27344	18858	21595	19821	21904	259162

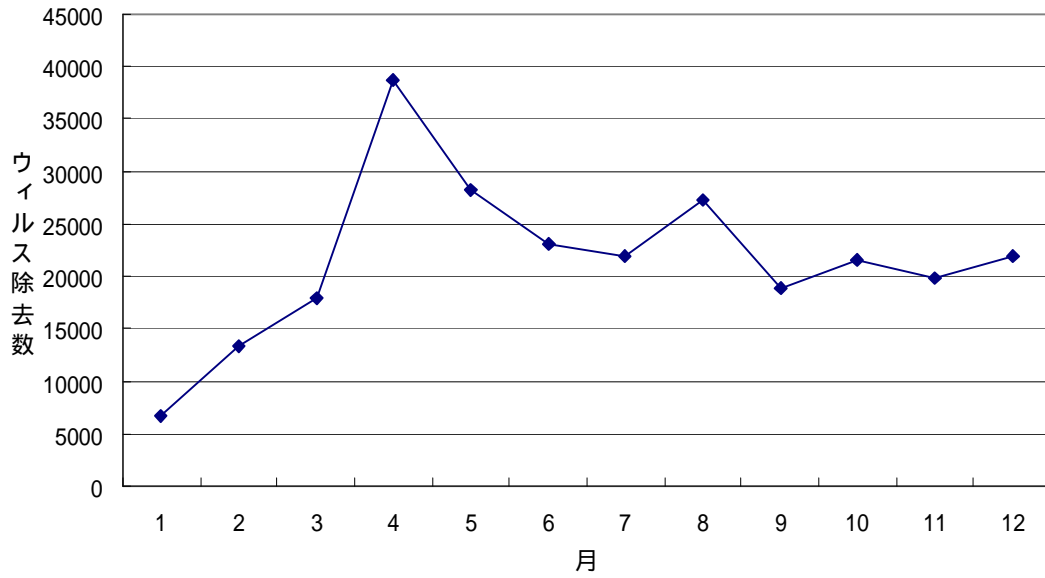


図1：ウイルス除去数全体の推移(2004/1~12)

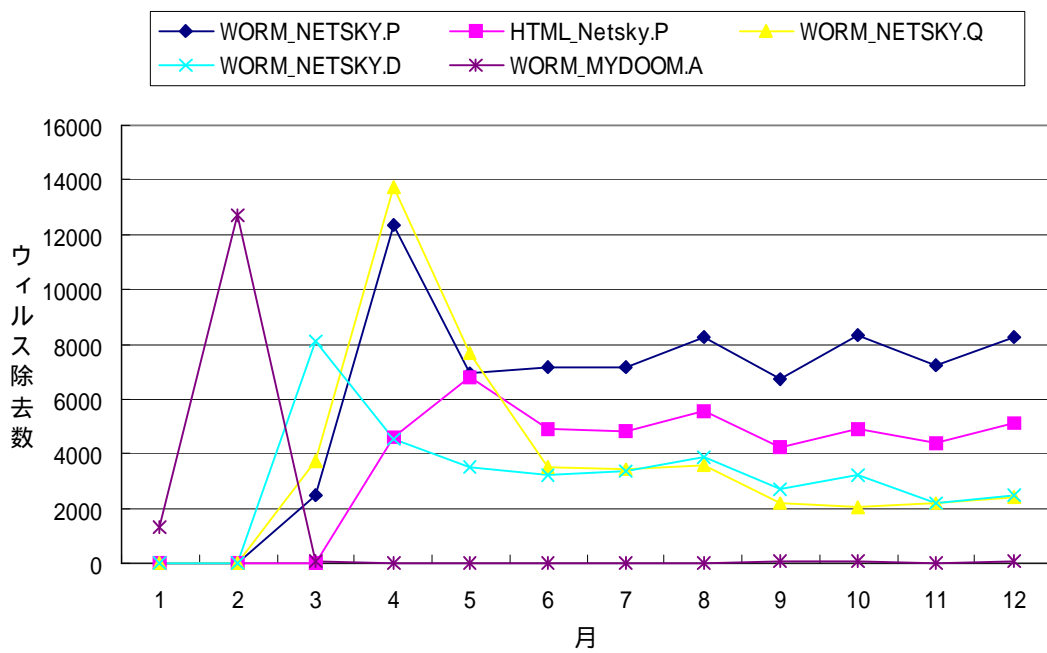


図2：ウイルス除去数ベスト10(1~5位)

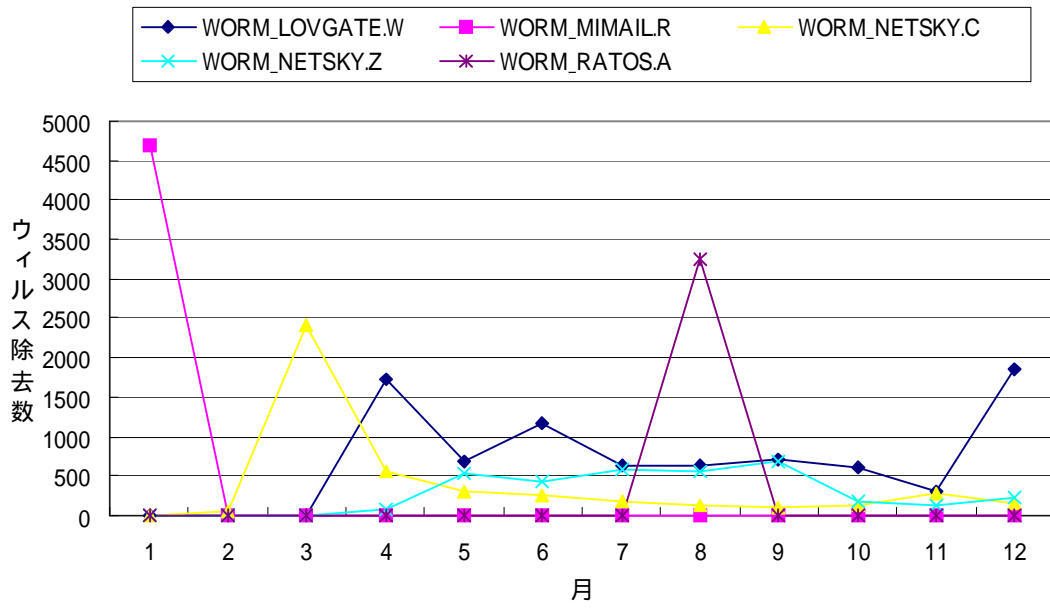


図3：ウィルス除去数ベスト10(6～10位)

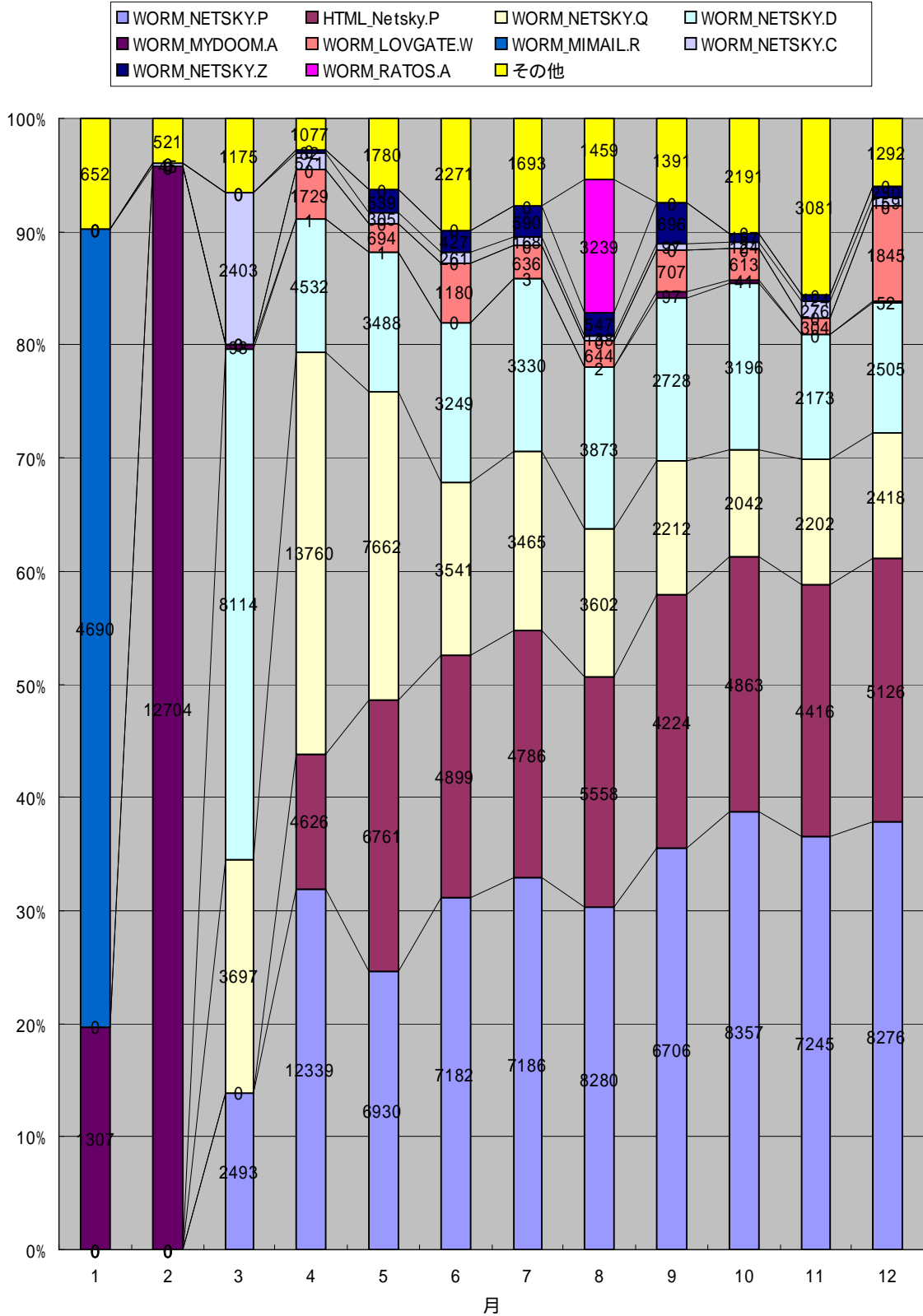


図4：各ウィルスの全体に占める割合（2004/1～12）

(b) 「その他」のウイルス

リスト1に示された「その他」に含まれるウイルスの内訳は次のリスト2の通りである。

リスト2：「その他」ウイルスの内訳

ウイルス名	削除数	ウイルス名	削除数	ウイルス名	削除数
WORM_BAGLE.Z	2051	HTML_IFRMEXP.GEN	74	WORM_BAGLE.GEN	6
WORM_BAGLE.AT	1607	PE_BUGBEAR.B-O	73	PE_Parite.A	5
WORM_NETSKY.J	1426	PE_BAGLE.N-O	70	WORM_BAGLE.F	5
WORM_KLEZ.H	1308	WORM_MYDOOM.T	70	WORM_BAGLE.V	5
WORM_BAGLE.GEN-1	827	WORM_NETSKY.W	61	WORM_HYBRIS.B	5
WORM_BUGBEAR.F	756	JS_ZEROLIN.A	55	WORM_MIMAIL.J	5
WORM_BAGLE.AU	697	PE_ZAFI.B	53	WORM_SOBER.G	5
WORM_MYDOOM.M	674	PE_BAGLE.P-O	50	Eicar_test_file	4
WORM_MYDOOM.F	626	PE_BUGBEAR.DAM	40	HTML_OBJECTEXP.A	4
WORM_MYDOOM.H	488	WORM_MIMAIL.E-1	38	PE_DUPATOR.1503	4
WORM_MYDOOM.L	365	WORM_NETSKY.T	38	PE_LOVGATE.AH-O	4
WORM_NETSKY.Y	365	WORM_SOBER.I	37	WORM_MIMAIL.G-1	4
VBS_BAGLE.Z	317	WORM_BAGLE.AD	36	WORM_MYDOOM.A	4
WORM_BAGLE.X	317	WORM_MYDOOM.S	36	WORM_NETSKY.AI	4
EMail_Flaw_MIME_Tag_Overflow	264	HTML_ZEROLIN.B	34	WORM_NETSKY.L	4
WORM_BAGLE.AG	260	WORM_MYDOOM.DAM	33	HTML_BOFRA.B	3
WORM_BAGLE.AM	247	WORM_NETSKY.R	31	HTML_ZEROLIN.C	3
WORM_Netsky.Dam	246	WORM_MIMAIL.C-1	28	JS_REDIRECT.A	3
WORM_MYDOOM.J	231	HTML_BAGLE.AC	25	JS_ZEROLIN.A-1	3
PE_FUNLOVE.4099	222	JS_ZEROLIN.D	24	WORM_BUGBEAR.A	3
WORM_BAGLE.AB	206	WORM_BAGLE.U	24	WORM_MIMAIL.F-1	3
WORM_MIMAIL.A	202	WORM_MIMAIL.I	24	JS_FORTNIGHT	2
PE_LOVGATE.AD	200	WORM_ZAFI.D	19	JS_RELINK.A	2
VBS_BAGLE.GEN	180	TROJ_CIDRA.D	18	WORM_BAGLE.D	2
WORM_BAGLE.AH	180	WORM_BAGLE.C	16	WORM_SWEN.B	2
WORM_NETSKY.X	170	WORM_BAGLE.E	16	X97M_LAROUX.XE	2
PE_LOVGATE.AC-O	169	WORM_MYDOOM.B	16	HTML_SMITFRAUD.A	1
WORM_NETSKY.AB	168	WORM_NETSKY.F	15	HTML_WAMUFRAUD.A	1
WORM_BAGLE.A	166	WORM_BUGBEAR.D	14	HTML_ZEROLIN.A	1
WORM_SWEN.A	149	HTML_CITIFRAUD.H	13	JOKE_GHOST.A	1
HTML_CITIFRAUD.C	137	HTML_MYDOOM.AH	13	PE_DUMARU.A	1
WORM_LOVGATE.Q	131	JS_ZEROLIN.E	13	PE_Magistr.B	1
WORM_BAGZ.D	127	VBS_REDLOF.A	13	VBS_INOR.U	1
VBS_BAGLE.X	121	WORM_BAGLE.K	13	WORM_BADTRANS.B	1
WORM_LOVGATE.R	119	HTML_BAGLE.AL	12	WORM_BAGLE.F-1	1
WORM_BAGLE.J	118	VBS_REDLOF.A-1	12	WORM_BLUEWORM.A	1
WORM_NETSKY.B	117	WORM_NETSKY.N	12	WORM_BLUEWORM.C	1
WORM_LOVGATE.V	114	WORM_SWASH.A	12	WORM_BUCHON.B	1
HTML_CONYC.A	113	WORM_LOVGATE.AF	11	WORM_DUMARU.Y	1
HTML_BAGLE.Z	109	JS_ZEROLIN.B	10	WORM_KLEZ.E	1
HTML_SUNFRAUD.B	109	PE_BAGLE.P-1	10	WORM_LOVGATE.AG	1
PE_VALLA.A	105	WORM_BAGLE.B	10	WORM_LOVGATE.F	1
WORM_FUNBAG.GEN	94	HTML_CITIFRAUD.E	8	WORM_LOVGATE.X	1
WORM_YAHA.G	93	PE_BAGLE.N-1	8	WORM_MIMAILS	1
PE_ELKERN.D	90	WORM_MABUTU.A	8	WORM_NETSKY.AC	1
WORM_LOVGATE.DAM	86	PE_SPACES.1445	7	WORM_NETSKY.C-1	1
WORM_NETSKY.M	84	VBS_LOVELORN.A	7	WORM_NETSKY.P	1
WORM_PLEXUS.B	84	WORM_LOVELORN.A	7	WORM_SIRCAM.A	1
WORM_NETSKY.S	79	HTML_BANKFRAUD.A	6	WORM_SOBER.F	1
WORM_BAGLE.AF	76	PE_TENROBOT.C	6	X97M_LAROUX.A	1

Apache James の設定と動作検証 (I)

—SMTP サーバとしての基本的な設定—

教育学部 (人間環境教育)

青木 昌三

■はじめに

本稿では、Apache Software Foundation(ASF) の James プロジェクトによるオープンソース電子メールサーバ Java Apache Mail Enterprise Server (別名 Apache James) の基本的な設定と動作検証を中心に紹介する。

Apache James は Java で書かれた電子メールアプリケーションのプラットフォームでもあり、SMTP サーバとしてのメール送受信の他、POP3 サーバ機能、NNTP 対応ニュースサーバ機能、FetchMail 機能、リモートからの Telnet によるメールサーバ管理機能等を備えている。その上、そうした諸機能に関わる設定もバラエティに富み、かつ容易である。

筆者が Apache James に関心をいだいたきっかけは、簡単な Java ベースでの Web アプリケーションの試作において、Java Mail の API を利用したことであった⁽¹⁾。文献(2)でも紹介されているように、Web アプリケーションの開発ではメールクライアント機能だけでなく、メールサーバ機能が必要となることが少なくない。そのサーバ機能が、例えば同じホスト内といった風に身近にあって、更に設定やカスタマイズが容易であれば、何かと便利である。Apache James はそのようなサーバに最も適したものの1つと云えよう⁽³⁾。

以下では、Apache James について SMTP サーバとしての基本的な機能に関わる各種の設定とその動作の検証について紹介するが、設定の実際は Apache James のドキュメンテーションに沿いつつ行う^(*)。設定内容等の多くは、これまで本年報で紹介してきたメール関連の記事の中で取り扱った事項と関連する^{(4) (5)}。是非、SMTP サーバプログラムである Sendmail や Qmail と比較していただきたい。尚、Apache James の概要や簡単な設定については、文献(2)、(3)を参照してほしい。

読者の Java 利用環境が様々であろうと考えられるので、Apache James のインストール等については、次の OS 上において行っている。

Windows 系： Windows2000(SP4)、WindowsXP(SP2)

Linux 系： Linux MLD 7

FreeBSD 系： FreeBSD-Release 4.9

(*1)<http://james.apache.org/>上の Documentation を参照。

■Java2SDK のインストール

Apache Jame-2.2.0 (以下では、簡単に Apache James と記す。)を利用するには Java2 の開発環境 SDK (Software Development Kit, Java2SDK) が必要で、インストールが未だの場合は新しいバージョンの SDK をダウンロードし、Apache James のインストールに

先立ってインストールしておく。Java2SDK のダウンロードは次の URL から行う。

<http://java.sun.com/j2se/>

Java2SDK のインストールや環境変数の設定については既に文献(6)に詳しく説明しているが、初めての読者もあろうかと思われるので、簡単に紹介する。

◆インストール

筆者の Windows2000 環境では既に j2sdk-1.4.2_03 がインストール済みであり、また FreeBSD 環境でも FreeBSD ネイティブな jdk がインストールされており、簡便のため出来るだけ設定済みの Java2SDK を用いることとした^(*)。

Windows2000/XP でのインストール

用いたバイナリ版 SDK は、

j2sdk-1_4_2_03-windows-i586-p.exe

である。インストールは、j2sdk-1_4_2_03-windows-i586-p.exe の実行で、指示に従うことにより行える。筆者の場合、インストール先ディレクトリを c:\¥j2sdk1.4.2_03 としている。

Linux でのインストール

用いたバイナリ版 SDK は

j2sdk-1_4_2_03-linux-i586.bin

である。j2sdk-1_4_2_03-linux-i586-rpm.bin (RedHat 系 Linux) を用いてもよい。

j2sdk-1_4_2_03-linux-i586.bin を用いる場合は、インストールするディレクトリ (例えば /usr/local/) にファイルをコピーした後インストール先で次のように実行する。

```
# sh j2sdk-1_4_2_03-linux-i586.bin
```

インストール先ディレクトリは、/usr/local/での展開だと、/usr/local/j2sdk1.4.2_03 となる。

j2sdk-1_4_2_03-linux-i586-rpm.bin を用いる場合は、適当なディレクトリでバイナリファイルを展開した後、rpm コマンドによりインストールが可能である (以下参照)。

```
# sh j2sdk-1_4_2_03-linux-i586-rpm.bin
# rpm -ivh j2sdk-1_4_2_03-linux-i586-rpm
```

FreeBSD でのインストール

文献(1)で紹介した Tomcat と Cocoon2 の利用において設定済みの FreeBSD ネイティブな j2sdk-1.4.2.01 環境をそのまま利用する^(*)、^(*)。インストール先ディレクトリは、/usr/local/jdk1.4.1 である。インストールの詳細については、文献(1)を参照してほしい。

(*1)本稿執筆時点の 2004 年 10 月現在の最新バージョンは、j2sdk-1.4.2_06, j2se5.0JDK である。新しいバージョンのものをインストールする場合には、本文中の該当箇所を適当に読みかえてほしい。

(*2)インストールは、/usr/ports/java/jdk14/において行ったのであるが、次の 3 つのファイルを自己の責任において予めダウンロードし、/usr/ports/distfiles/内に置いておくことが必要であった。

パッチセットファイル→ bsd-jdk14-patches-3.tar.gz

ソースコードファイル→ j2sdk-1_4_1-src-scs1.zip

Linux Java 用の j2sdk ファイル→j2sdk-1_4_2_01-linux-i586.bin

インストールの実行は、

```
# cd /usr/ports/java/14
# make
# make install
```

である。インストール先ディレクトリは/usr/local/jdk1.4.1 であり、Linux Java も同時に /usr/local/linux-sun-jdk1.4.2_01/ にインストールされる。

(*3)Linux 用の j2sdk-1_4_2_03-linux-i586.bin も、/usr/local/j2dk1.4.2_03 にインストールされている。インストールの方法は、上述の Linux の場合と同様である。FreeBSD で Linux 版の j2sdk を用いるには、Linux バイナリ互換機能の設定が必要である。この互換機能の設定についても、文献(1)を参照してほしい。

◆環境変数の設定

環境変数 JAVA_HOME に Java2SDK のインストール先ディレクトリを設定する。

Windows2000/XP の場合

「コントロールパネル」→「システム」→「詳細」→「環境変数」により設定する。

B シェル系の場合 (例)

JAVA_HOME=/usr/java/j2sdk1.4.2_03 ; export JAVA_HOME

C シェル系の場合 (例)

setenv JAVA_HOME /usr/local/jdk1.4.1

■Apache James のインストール

◆Apache James の入手

Apache James のダウンロード元の URL は次の通りである。

<http://james.apache.org/download.cgi>

各 OS によって、それぞれ次の Release Builds のバイナリ版を入手した(*1)。

Windows の場合： james-2.2.0.zip ファイル

Linux, FreeBSD の場合： james-2.2.0.tar.gz ファイル

Apache James は 100% Java(Pure Java)で書かれたアプリケーションであり、Windows 用、UNIX 用と分かれているわけではないので、上記の2つのバイナリ版の利用は実は OS によらない。2つのファイルの利用は、単に(筆者の)インストールの簡便さに基づいている。

(*1)本稿執筆時点(2004.10)での最新バージョンは 2.2.0 である。異なるバージョンのものを利用する場合には、本文中のファイル名等は、実際にインストールしたバージョンのものに読みかえてほしい。

◆Apache James のインストール

Windows2000/XP でのインストール

james-2.2.0.zip ファイルを適当なフォルダ内で展開すると、james-2.2.0 フォルダが作成される。インストールはそれだけである。利用上の便宜のため、このフォルダをルートフォルダにコピーし、c:\james-2.2.0 とした。

Linux, FreeBSD でのインストール

james-2.2.0.tar.gz ファイルをインストール先のディレクトリ(例えば、/usr/local/)にコピーし、以下のように tar コマンドで展開する。インストール先ディレクトリは、例えば、/usr/local/james-2.2.0 となる(以下参照)。

```
sl-epson# tar zvxf tomcat-4.1.24.tar.gz ← 解凍・展開
sl-epson# cd james-2.2.0
sl-epson# ls
LICENSE.txt      apps             docs             logs
README           bin             ext             temp
```

README.txt	conf	lib	work
sl-epson#			

◆Apache James の起動と終了

何よりもまず、Apache James の起動と終了を確認する。初期設定で、Remote Manager Service, POP3 Service, SMTP Service, NNTP Service の4つのサービスが有効となる。

Windows2000/XP での起動と終了

コマンドプロンプト画面で、c:\james-2.2.0\bin フォルダに移動し、run.bat を実行する。以下の様であれば、最初の関門はパスである^(*1)。尚、WindowsXP で、ipv6 を有効としているとエラーが生じるので注意が必要である^(*2)、^(*3)。

```
C:\james-2.2.0>cd bin
C:\james-2.2.0\bin>run
Using PHOENIX_HOME: C:\james-2.2.0
Using PHOENIX_TMPDIR: C:\james-2.2.0\temp
Using JAVA_HOME: c:\j2sdk1.4.2_03

Phoenix 4.0.1

James 2.2.0
Remote Manager Service started plain:4555
POP3 Service started plain:110
SMTP Service started plain:25
NNTP Service started plain:119
Fetch POP Disabled
FetchMail Disabled
```

次いで、Ctrl+C キーを押して、プロセスを中断する。

```
JVM exiting abnormally. Shutting down Phoenix.
バッチ ジョブを終了しますか (Y/N)? y
```

```
C:\james-2.2.0\bin>
```

Linux, FreeBSD での起動と終了

/usr/local/james-2.2.0/bin/run.sh を実行する (FreeBSD の場合、以下参照)。Linux の場合も同様である^(*4)、^(*5)。

```
sl-epson# cd james-2.2.0
sl-epson# cd bin
sl-epson# printenv          ← 環境変数を確認する
TERM=vt100
PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/games:/usr/local/sbin:/usr/local/bin:/usr
===== 中略 =====
JAVA_HOME=/usr/local/jdk1.4.1      ← JAVA_HOME が設定されている
CATALINA_HOME=/usr/local/jakarta-tomcat-4.1.29
sl-epson# ./run.sh
./run.sh: Permission denied.      ← 実行属性が設定されていない
sl-epson# chmod 555 run.sh
sl-epson# ./run.sh
./run.sh: /usr/local/james-2.2.0/bin/phenix.sh: permission denied
sl-epson# chmod 555 phoenix.sh    ← phoenix.sh に実行属性を設定
sl-epson# ls -l                  ← 実行属性の確認
total 192
-rw-r--r-- 1 root wheel 49152 Jun 16 13:30 Wrapper.dll
```

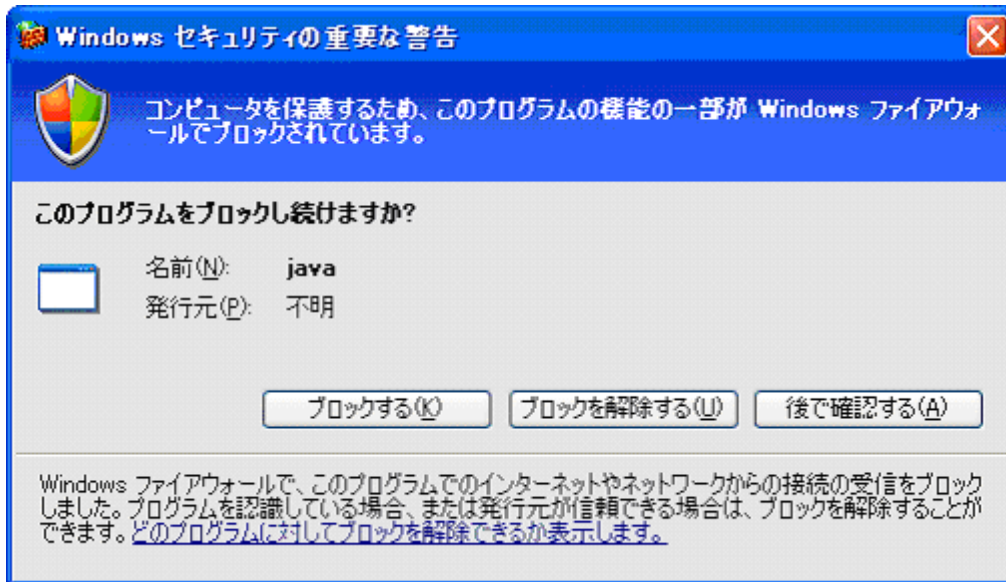
```

-rw-r--r-- 1 root wheel 98304 Jun 16 13:30 Wrapper.exe
drwxr-xr-x 2 root wheel 512 Oct 10 09:33 lib
-rw-r--r-- 1 root wheel 7572 Jun 16 13:30 phoenix-loader.jar
-rwxr-xr-x 1 root wheel 6162 Jun 16 13:30 phoenix.sh
-rw-r--r-- 1 root wheel 2643 Jun 16 13:30 run.bat
-r-xr-xr-x 1 root wheel 854 Jun 16 13:30 run.sh
-rw-r--r-- 1 root wheel 24290 Jun 16 13:30 wrapper.jar
sl-epson# ./run.sh
Using PHOENIX_HOME: /usr/local/james-2.2.0
Using PHOENIX_TMPDIR: /usr/local/james-2.2.0/temp
Using JAVA_HOME: /usr/local/jdk1.4.1
Running Phoenix:

Phoenix 4.0.1

James 2.2.0
Remote Manager Service started plain:4555
POP3 Service started plain:110
SMTP Service started plain:25
NNTP Service started plain:119
Fetch POP Disabled
FetchMail Disabled
^C ← ^C を入力
JVM exiting abnormally. Shutting down Phoenix.
sl-epson#
    
```

(*1)WindowsXP(SP2)のデフォルト設定ではセキュリティ機能が有効となっている。そのため、初めての起動では次のような画面が現れる。「ブロックを解除する」または「後で確認する」を選択する。



(*2)ipv6 を有効としていると、ipv6 アドレスが理解不能としてエラーとなるようで、起動に失敗する(以下参照)。

```

C:\james-2.2.0\bin>ipv6 install
Installing...
Succeeded.

C:\james-2.2.0\bin>run
Using PHOENIX_HOME: C:\james-2.2.0
Using PHOENIX_TMPDIR: C:\james-2.2.0\temp
    
```



```
Using JAVA_HOME:      c:\j2sdk1.4.2_03
```

```
Phoenix 4.0.1
```

```
===== 中略 =====
```

```
-----
The log file may contain further details of error.
Please check the configuration files and restart Phoenix.
If the problem persists, contact the Avalon project.  See
http://jakarta.apache.org/avalon for more information.
Shutting down Phoenix.
C:\james-2.2.0\bin>
```

例えば、C:\james-2.2.0\apps\james\logs フォルダ内の DNS サーバに関わるログに次のような記述が見られる。

```
25/10/04 05:46:00 INFO  dnsserver: Autodiscovery is enabled - trying to discover your system's DNS
Servers
25/10/04 05:46:02 INFO  dnsserver: Adding autodiscovered server 192.168.0.1
25/10/04 05:46:02 INFO  dnsserver: Adding autodiscovered server fec0:0:0:ffff::1%1 ←ipv6 アドレス
に注目
25/10/04 05:46:02 INFO  dnsserver: DNS Server is: 192.168.0.1
25/10/04 05:46:02 INFO  dnsserver: DNS Server is: fec0:0:0:ffff::1%1 ←ipv6 アドレスに注目
25/10/04 05:46:02 FATAL dnsserver: DNS service could not be initialized.  The DNS servers specified
are not recognized hosts.
java.net.UnknownHostException: fec0:0:0:ffff::1%1: fec0:0:0:ffff::1%1 ←ipv6 アドレスに注目
        at java.net.InetAddress.getAllByName0(InetAddress.java:1011)
===== 以下 略 =====
```

(*3) WindowsNT/2000/XP 上での実際の運用にあたっては、サービスとして組み込めばより好都合である。Ver.2.1 より wrapper.exe プログラムが提供されているので、サービスとしてインストールするにはこれを用いる。まず c:\james-2.2.0\bin ディレクトリにおいて、コンソールアプリケーションとして起動するよう、次のように実行してみる。

```
C:\james-2.2.0\bin>wrapper -c ../conf/wrapper.conf
```

これでうまく Apache James が起動すればサービスとしての組み込みも可能であるので、上記の“-c”を“-i”に替えて実行すればよい。尚このとき、Java Runtime Environment バージョンのレジストリへの登録が 1.4 でないと起動に失敗するので注意が必要である (以下参照)。

```
C:\james-2.2.0\bin>wrapper -c ../conf/wrapper.conf
wrapper | --> Wrapper Started as Console
wrapper | Launching a JVM...
jvm 1   | Registry key 'Software\JavaSoft\Java Runtime Environment\CurrentVersion'
jvm 1   | has value '1.3', but '1.4' is required.
jvm 1   | Error: could not find java.dll
jvm 1   | Error: could not find Java 2 Runtime Environment.
===== 中 略 =====
wrapper | <-- Wrapper Stopped
```

(*4) 当然のことであるが、sendmail を起動してはならない。sendmail が起動していると、ポートが既に利用されているとしてエラーとなる。ipv6 については、ipv6 を enable としていても、WindowsXP とは異なり、起動ではエラーを生じていない。

(*5) FreeBSD 環境にインストールされている Linux 用 j2sdk-1_4_2_03-linux-i586.bin を用いても (Linux バイナリ互換機能を有効)、FreeBSD ネイティブ Java と同様の起動・終了の結果が得られる (以下参照)。

```
sl-epson# setenv JAVA_HOME /usr/local/j2sdk1.4.2_03
sl-epson# printenv | grep JAVA
JAVA_HOME=/usr/local/j2sdk1.4.2_03
sl-epson# cd james-2.2.0/bin
sl-epson# ./run.sh
Using PHOENIX_HOME:  /usr/local/james-2.2.0
```

```

Using PHOENIX_TMPDIR: /usr/local/james-2.2.0/temp
Using JAVA_HOME:      /usr/local/j2sdk1.4.2_03
Running Phoenix:

Phoenix 4.0.1

James 2.2.0
Remote Manager Service started plain:4555
POP3 Service started plain:110
SMTP Service started plain:25
NNTP Service started plain:119
Fetch POP Disabled
FetchMail Disabled

^CJVM exiting abnormally. Shutting down Phoenix.
sl-epson#

```

■ Apache James の基本動作確認

Apache James の起動と終了について確認したが、更にメール送受信機能等が正常に動作するかどうかの確認が必要である。

◆ テスト環境の構成

Apache James の初期設定、ユーザ登録、メール受信など一連のテストを行うにあたっての、各ホストマシンの構成を示しておこう (図1参照)。

Windows2000(SP2)ホスト： ホスト名 end-pro-1000 IP アドレス 192.168.0.2

Linux MLD 7 ホスト： ホスト名 end-pro-1000 IP アドレス 192.168.0.2

WindowsXP(SP2)ホスト： ホスト名 ravie01 IP アドレス 192.168.0.4

FreeBSD-Release 4.9 ホスト： ホスト名 sl-epson IP アドレス 192.168.0.8

ドメイン名 (仮)： aoki-home.kagawa-u.ac.jp

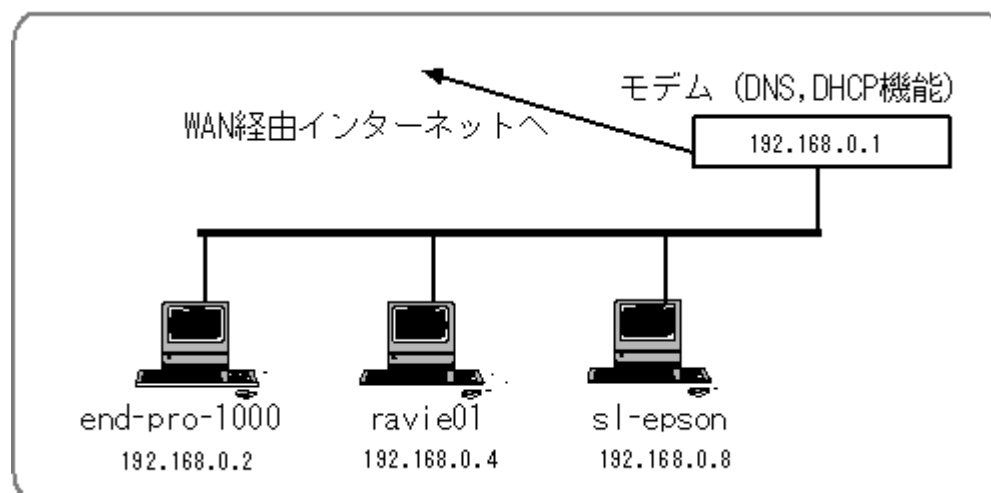


図1：テスト環境のホスト構成

Windows2000(SP2)とLinux MLD7のOSはend-pro-1000にインストールされており、それぞれ切り替えてブートし利用する。3台のホストマシンはDNS機能^(*)、DHCP機能を有するモデム(IPアドレス192.168.0.1)に接続され、WANを経由して香川大学総合情報基盤センター幸町分室或いはネットワークプロバイダーのPPPサーバに接続、インターネット

トへと繋がっている。

後の説明の都合上、各ホストマシンの DNS サーバ設定について紹介しておこう。各ホストでは、現時点では次の様となっている。

Windows2000(SP2)ホスト： TCP/IP のプロパティ設定で、「DNS サーバのアドレスを自動的に取得する」としている。従って、DNS サーバについての情報は、図1に示されているモデムから得る。

Linux MLD 7 ホスト： Linux MLD 7では、同じホストマシンにインストールされている Windows2000 ホストの設定から各種情報を得る。DNS サーバとして、Windows2000 環境と同じに、モデム(192.168.0.1)が設定されている（以下参照）。

```
[aoki@end-pro-1000 etc]$ cat resolv.conf
; generated by /sbin/dhclient-script
search aoki-home.kagawa-u.ac.jp
nameserver 192.168.0.1
[aoki@end-pro-1000 etc]$ cat host.conf
order hosts,bind
[aoki@end-pro-1000 etc]$
```

WindowsXP(SP2)ホスト： Windows2000 ホストと同様の設定である。

FreeBSD-Release 4.9 ホスト： DNS サーバはローカルホスト(127.0.0.1)とモデム(192.168.0.1)に設定。resolv.conf, host.conf の内容を示す。

```
%cd /etc
%cat resolv.conf
domain aoki-home.kagawa-u.ac.jp
nameserver 127.0.0.1
nameserver 192.168.0.1
%cat host.conf
# $FreeBSD: src/etc/host.conf,v 1.6 1999/08/27 23:23:41 peter Exp $
# First try the /etc/hosts file
hosts
# Now try the nameserver next.
bind
# If you have YP/NIS configured, uncomment the next line
# nis
%
```

上記の各ホストでの設定からも分かるように、多くは DNS サーバとして 192.168.0.1 アドレスのモデムを指定している(*1)。そのため、各ホストの hosts ファイルにテスト環境下の全てのホストマシンについての情報を記述し、名前解決に当たっては hosts ファイル情報が優先されるようにしている。

(*1)ここでの DNS 機能とは、モデムが接続先の PPP サーバ等を介して DNS に関わる情報を得ることであり、モデム自身がゾーンファイル等を有する DNS サーバ機能そのものを有しているのではない。

◆初期設定

James ドキュメントに従い、初期設定を行う。設定の中心となる設定ファイルは、%JAMES_HOME%/apps/james/SAR_INF/config.xml ファイルである。ここで、%JAMES_HOME%は Apache James のインストール先ディレクトリを表している。

config.xml ファイルはインストールを行っただけでは展開されず存在しない。まず Apache James を起動し、そして一旦停止することが必要である⁽²⁾。ドキュメントによれば、

1) 起動, 2) 停止, 3) 設定ファイル編集, 4) 再起動

の順序である。一旦停止することにより config.xml ファイルが展開され、編集が可能となる^(*)。

インストール後に先ず行うべき設定は次の4点である。

◆リモートマネージャ管理者パスワードの設定

Telnet を用いて遠隔からユーザ登録や削除、Apache James の終了など、メールサーバの管理を行う機能を提供するのがリモートマネージャである。ここでは、リモートマネージャ管理者のパスワードを設定する。初期設定でのパスワードは”root”と設定されているので、この部分を変更する (config.xml の一部抜粋、以下参照)。

```
<administrator_accounts>
<!-- CHECKME! -->
    <!-- Change the default login/password. -->
    <account login="root" password="*****"/>    ←パスワード設定
</administrator_accounts>
```

◆DNS サーバのアドレス設定

DNS サーバについて、利用環境における DNS サーバに設定する。初期設定は次のようになっている (config.xml の該当部分、以下参照)

```
<dnsserver>
  <servers>
    <!--Enter ip address of your DNS server, one IP address per server -->
    <!-- element. -->
    <!--
      <server>127.0.0.1</server>
    -->
  </servers>
  <!-- Change autodiscover to false if you would like to turn off autodiscovery -->
  <!-- and set the DNS servers manually in the <servers> section -->
  <autodiscover>>true</autodiscover>
  <authoritative>>false</authoritative>
</dnsserver>
```

利用環境での DNS サーバを具体的に設定する場合は、上記の<server>タグ内に設定し (2つ以上でも構わない)、<autodiscover>>false</autodiscover>とする (次の例参照)。

```
<dnsserver>
  <servers>
    <server>133.92.6.17</server>    ←   サーバのアドレス設定
    <server>133.92.6.18</server>    ←   サーバのアドレス設定
  </servers>
  <autodiscover>>false</autodiscover>    ←   false に設定
  <authoritative>>false</authoritative>
</dnsserver>
```

ここでは、テスト環境下でもあり、<autodiscover>>true</autodiscover>となっているので、とりあえず初期設定のままとする^(*) ⁽²⁾ ⁽³⁾。

◆メールサーバでローカルとして取り扱うドメイン名/IP アドレスの設定

デフォルトでは, Apache James は localhost に登録されたユーザ宛のメールしか取り扱わない。ここでは, Apache James でローカルなものとして取り扱い受信するドメイン名または IP アドレスを設定する。Sendmail での/etc/mail/local-host-names ファイルに記述する内容と同じ設定である。

初期設定に加えて, 例えば, 次の様に設定しておこう (ravie01 ホストの場合, 関連部分を抜粋, 以下参照) (*4) (*5)。

```
<servername autodetect="true" autodetectIP="true">
<!-- CONFIRM? -->
    <servername>localhost</servername>    ← 初期設定
    <servername>aoki-home.kagawa-u.ac.jp</servername> ← 追加設定(*4)
<servername>ravie01.aoki-home.kagawa-u.ac.jp</servername> ← 追加設定(*5)
</servername>
```

◆Postmaster のアドレスの設定

エラー等の送信先としての Postmaster のメールアドレスを設定する。とりあえず, 初期設定のままとする (関連部分抜粋, 以下参照) (*6)。

```
<!-- CHECKME! -->
<!-- This is the postmaster email address for this mail server. -->
<!-- Set this to the appropriate email address for error reports -->
<!-- If this is set to a non-local email address, the mail server -->
<!-- will still function, but will generate a warning on startup. -->
<postmaster>Postmaster@localhost</postmaster>
```

(*1) 「一旦停止する」は, 後述するリモートマネージャに Telnet で接続し, shutdown コマンドを用いるのが正しいようである。^C の押下により一旦プログラムを停止し, james.sar ファイルの展開により作成された%JAMES_HOME%/apps/james/SAR_INF/config.xml ファイルを編集する方法は, WindowsXP, Linux, FreeBSD 各ホストでは問題がなかったが, Windows2000 ホストでは SAR-INF フォルダが消滅する (従って config.xml ファイルも消滅する), conf フォルダは存在するがその中身のファイル等は消滅するといったことが生じた。

Windows2000 で Apache James を停止せずに config.xml ファイルを編集し, その後に ^C で停止する方法を採ってみると, 一旦編集された config.xml ファイルは消滅せず, 以後 Apache James は再起動により動作した。ただこのときも, 他の OS の場合に存在する SAR-INF フォルダ内の他のファイルや conf フォルダ内のファイル等は一旦停止により消滅してしまった。最初の config.xml ファイルの編集時にリモートマネージャの shutdown コマンドにより停止した場合には, 以後は ^C により停止しても全てのファイルが消滅せずそのままに, 他の OS と変わらない状況で動作している。

(*2) Apache James 起動時に作成される DNS サーバについての log ファイル (%JAMES_HOME%/apps/james/logs 内)によれば, Windows2000 ホストでは次のように設定されている (WindowsXP, Linux ホストでも同様)。

```
10/10/04 08:13:35 INFO dnsserver: Autodiscovery is enabled - trying to discover your system's DNS Servers
10/10/04 08:13:35 INFO dnsserver: Adding autodiscovered server 192.168.0.1
10/10/04 08:13:35 INFO dnsserver: DNS Server is: 192.168.0.1
```

FreeBSD ホストでは

```
25/10/04 05:09:06 INFO dnsserver: Autodiscovery is enabled - trying to discover your system's DNS Servers
25/10/04 05:09:06 INFO dnsserver: Adding autodiscovered server 127.0.0.1
25/10/04 05:09:06 INFO dnsserver: Adding autodiscovered server 192.168.0.1
25/10/04 05:09:06 INFO dnsserver: DNS Server is: 127.0.0.1
25/10/04 05:09:06 INFO dnsserver: DNS Server is: 192.168.0.1
```

である。上述の各ホストにおける DNS 設定がそのまま反映されていることに注意してほしい。

(*3) 香川大学情報総合基盤センター幸町分室内で用いている WindowsXP ホストでは, DNS サーバのアドレスが固定で設定されている。この場合, config.xml での初期設定で次の様に設定どおりの DNS

サーバアドレスが得られている。

```
13/10/04 16:15:48 INFO dnsserver: Autodiscovery is enabled - trying to discover your system's DNS Servers
13/10/04 16:15:48 INFO dnsserver: Adding autodiscovered server 133.92.6.17
13/10/04 16:15:48 INFO dnsserver: DNS Server is: 133.92.6.17
```

(*4)実際の運用では、aoki-home.kagawa-u.ac.jp が当該ホストを指すように、DNS サーバに MX レコードとしての登録が必要である。

(*5)ravie01 の部分は、それぞれのホスト名で置き換える。追加設定の必要性については、後述する「●デフォルト設定での自ホスト宛メールの受信とメール中継◆自ホスト宛メールの受信」の(*1)を参照のこと。

(*6)実際の運用にあたっては、postmaster ユーザを登録し、postmaster 宛メールが転送されるよう alias の設定を行うべきである。

◆テストユーザの登録

ユーザの登録は、Telnet を用いてリモートマネージャにより行う。このためのポート番号は、Apache James 起動時に表示されるように、初期設定では 4555 番である（他のポート番号に変更可能）。リモートマネージャでは、ユーザ登録機能に加え、次のような管理コマンドが提供されている（表 1 参照）。

表 1：リモートマネージャにおける管理コマンド

コマンド	機能
help	ヘルプを表示（コマンド一覧を表示）
listusers	登録ユーザの一覧を表示
countusers	登録ユーザ数を表示
adduser [username] [password]	新規ユーザを登録
verify [username]	特定ユーザの存在を確認
deluser [username]	登録ユーザを削除
setpassword [username] [password]	ユーザのパスワードを設定
setalias [user] [alias]	エイリアスを設定（user 宛メールを alias 名宛にローカル転送）
showalias [username]	ユーザのエイリアス設定を表示
unsetalias [user]	ユーザのエイリアス設定を解除
setforwarding [username] [emailaddress]	ユーザ宛メールを他のアドレスに転送
showforwarding [username]	ユーザのメール転送設定を表示
unsetforwarding [username]	ユーザのメール転送設定を解除
user [repositoryname]	ユーザリポジトリを変更
shutdown	Apache James を終了
quit	コネクションを切断

テストユーザとして s-user01, r-user01 を登録する。4555 番ポートを指定し localhost 或は各ホストに Telnet で接続して、adduser コマンドを実行する（以下参照）。

```
JAMES Remote Administration Tool 2.2.0
Please enter your login and password
Login id: ← root を入力（表示されない）
Password: ← パスワードを入力（表示されない）
```

```

Welcome root. HELP for a list of commands
adduser s-user01 s-user01 ← 登録コマンドを入力
User s-user01 added
adduser r-user01 r-user01 ← 登録コマンドを入力
User r-user01 added
listusers ← 登録者一覧のコマンドを入力
Existing accounts 2
user: s-user01
user: r-user01

```

登録ユーザ情報の収納先は、%JAMES_HOME%/apps/james/var/users/である。

◆ デフォルト設定での自ホスト宛メールの受信とメール中継

先に登録したユーザ宛にメールを送信し、また受信して、デフォルト設定での基本的なメール機能を確認する。デフォルト設定で既に SMTP サーバ機能, POP3 サーバ機能が enable となっている。

● 自ホスト宛メールの受信

SMTP サーバは、自ホスト内の登録ユーザ宛メールはデフォルト設定で受信する。ここでは、end-pro-1000 ホスト (Windows2000) からメーラを用いて、end-pro-1000 自身, sl-spon (FreeBSD), ravie01 (WindowsXP) に登録されたユーザ r-user01 にメールを送信する。送信者名は s-user01 である。メーラでの SMTP サーバ名の設定は、それぞれメール送信先のホストとする(*1)。メールは、それぞれのホストマシンで受け付けられた。sl-epson ホストで受信されたメールを示す (以下参照)。

```

Return-Path: <s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp>
Received: from 192.168.0.2 ([192.168.0.2])
    by sl-epson.aoki-home.kagawa-u.ac.jp (JAMES SMTP Server 2.2.0) with
    SMTP ID 268
    for <r-user01@sl-epson.aoki-home.kagawa-u.ac.jp>;
    Sat, 30 Oct 2004 11:14:16 +0900 (JST)
Message-Id: <200410300154.AA00004@end-pro-1000.ed.kagawa-u.ac.jp>
From: S-USER01 <s-user01@end-pro.aoki-home.kagawa-u.ac.jp>
Date: Sat, 30 Oct 2004 10:54:47 +0900
To: r-user01@sl-epson.aoki-home.kagawa-u.ac.jp
Subject: テストメール 0 1
===== 中 略 =====
Delivered-To: r-user01@sl-epson.aoki-home.kagawa-u.ac.jp

```

テストメール
2004.10.30

user 名@aoki-home.kagawa-u.ac.jp 宛メールの受信

各ホストの設定で、<servername>aoki-home.kagawa-u.ac.jp</servername>を設定している。r-user01@aoki-home.kagawa-u.ac.jp 宛のメールが受信されることを確認する。ravie01 ホストで受信されたメールを示す。

```

Return-Path: <s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp>
Received: from end-pro-1000.aoki-home.kagawa-u.ac.jp ([192.168.0.2])
    by ravie01 (JAMES SMTP Server 2.2.0) with SMTP ID 353
    for <r-user01@aoki-home.kagawa-u.ac.jp>; ← 受信者名に注目
    Sat, 30 Oct 2004 13:38:13 +0900 (JST)
Message-Id: <200410300433.AA00010@end-pro-1000.ed.kagawa-u.ac.jp>
Date: Sat, 30 Oct 2004 13:33:31 +0900

```



```

To: r-user01@aoki-home.kagawa-u.ac.jp
Subject: テストメール03
From: S-USER01 <s-user01@end-pro.aoki-home.kagawa-u.ac.jp>
===== 中 略 =====
Delivered-To: r-user01@aoki-home.kagawa-u.ac.jp      ← 受信者名に注目

テストメール03
2004.10.30
    
```

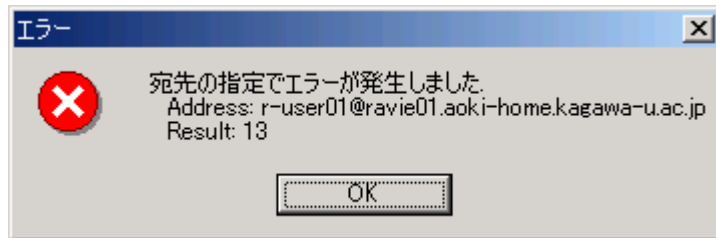
(*1)Windows ホストの場合、ホスト名の設定でドメイン名まで含めた設定をスキップしがちである。この場合、

```
<servername>localhost</servername>
```

の設定のみで

```
<servername>ravie01.aoki-home.kagawa-u.ac.jp</servername>
```

のようなフルホスト名の設定が無いとする。このとき、r-user01@ravie01 宛のメールは受け付けられるが (localhost=ravie01)、宛先のメールアドレスが r-user01@ravie01.aoki-home.kagawa-u.ac.jp であるメールは Apache James 側で受け付けられず、エラーとなる (下図参照)。



Windows と違って FreeBSD や Linux の場合には、hostname コマンドで sl-epson.aoki-home.kagawa-u.ac.jp

のようにドメイン名をつけたホスト名が得られるので、

```
<servername>sl-spson.aoki-home.kagawa-u.ac.jp</servername>
```

のような設定がなくても、メールは受け付けられる。それぞれのホスト上で hostname コマンドを実行し、得られる結果を参考とすればよい。

●メールの中継－初期設定での中継拒否－

SMTP サーバは、デフォルト設定では一般にメールを中継しないことが求められる。

end-pro-1000 ホスト内のメーラを用いて、SMTP サーバを ravie01 ホストに設定し、sl-epson ホストの r-user01 ユーザ宛にメールを送信する (図2 参照)。

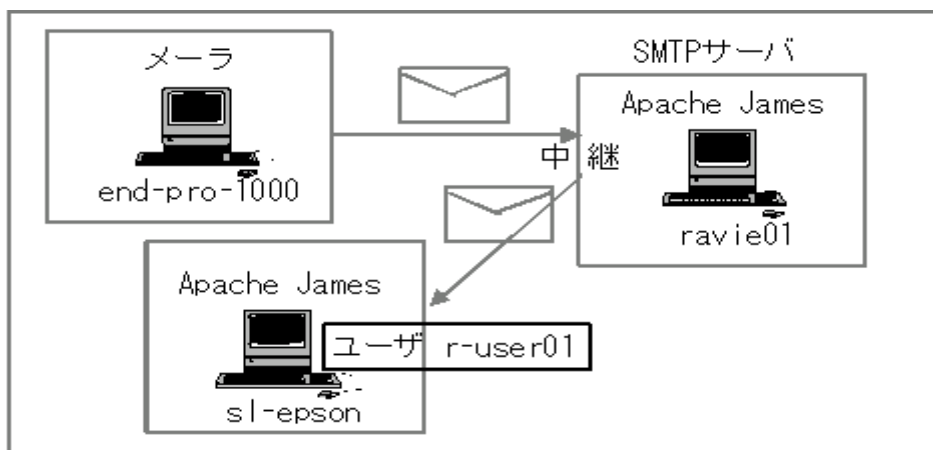


図2：メール中継のテスト

メールは中継されずエラーが表示された (図3 参照)。

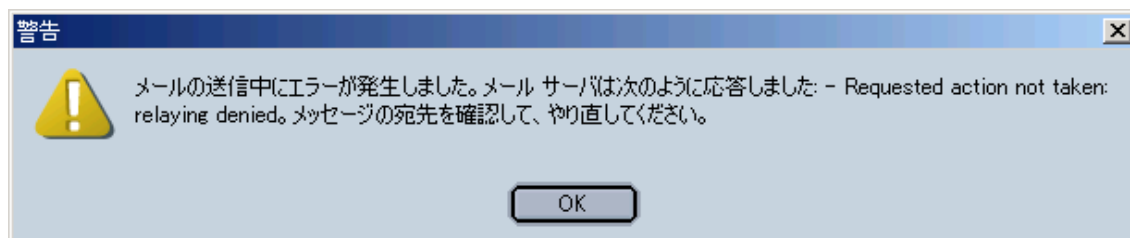


図 3：メール中継で生じたエラー表示

■ Apache James コンフィギュレーションの概要

Apache James でのコンフィギュレーションの概要を紹介する。コンフィギュレーションは、config.xml ファイルを編集して行う。

◆ config.xml ファイルの構造

config.xml は先ず幾つかの大きなブロック要素で構成されている（表 2 参照）。

表 2：config.xml を構成するブロック

ブロック	説明
James ブロック	Postmaster のアドレスやサーバ名等の設定
Fetch pop ブロック	Fetch POP 機能に関する諸設定
Spool Manager ブロック	メールの処理過程についての設定
DNS Server ブロック	DNS サーバに関わる情報の設定
Remotemanager ブロック	リモートマネージャについての設定
POP3 server ブロック	POP3 サーバ機能に関する設定
SMTP server ブロック	SMTP サーバ機能に関する設定
NNTP server ブロック	NNTP サーバ機能に関する設定
nntp-repository	NNTP で扱うデータのリポジトリについての設定
Mailstore ブロック	メールデータを収納するリポジトリについて設定
User Storage ブロック	ユーザ情報を収納するファイル等についての設定
Database-Connections ブロック	データベースやそれとの接続についての設定
Object Storage ブロック	ファイルリポジトリ等の詳細についての設定
Connection Manager ブロック	コネクション数, Idle-Timeout などの設定
Socket Manager ブロック	接続に用いる SSL ソケットなどについての設定
Thread Manager ブロック	スレッド数などのスレッドについての設定

先に初期設定のところで取り扱った項目はそれぞれ、

リモートマネージャ管理者パスワードの設定 → Remotemanager ブロック

DNS サーバのアドレス設定 → DNS Server ブロック

メールサーバでローカルとして取り扱うドメイン名/IP アドレスの設定と Postmaster のアドレスの設定 → James ブロック

に属している。メールの受信や中継等についての設定を取り扱うのは、Spool Manager ブロックにおいてである。

◆Spool Manager ブロックでのメール取り扱いの流れ

Spool Manager ブロックでは, Spool 上のメッセージ処理についての設定がとり扱われる。ブロックは, Spool の threads 数を決める<threads>要素 (規定値は 10), Maillet と Matcher プログラムをロードするパッケージ名定義を与える<mailetpackages>要素と<matcherpackages>要素, それに名前のついた幾つかの<processor>要素から構成されている。

●メール処理の流れを設定する<processor>要素

1つの<processor>要素は, Spool 上のメールを処理する単位 (ブロック) を構成し, メール処理は, それぞれ名前の付いた複数の<processor>要素ブロックを経て行われる。config.xml ファイルにデフォルトで記述されている<processor>要素ブロックを紹介しておこう (コメント扱いのものも含む, 表3参照)。

表3 : config.xml に記述されている<processor>要素

Processor 要素名	説明
root	メール処理の出発点となるプロセッサ (必須)
error	エラーメールを処理するプロセッサ (必須)
transport	ローカル宛, リモート宛メールを取り扱うプロセッサ
spam	スパムメールを取り扱うプロセッサ
local-address-error	登録されていないユーザ宛メール等を処理するプロセッサ
relay-denied	中継を拒否したメールを取り扱うプロセッサ

config.xml ファイルに記述されているプロセッサのブロックは表3の通りであるが, 勿論独自に<processor>要素を定義しても一向に差し支えない。その際には, 上記の<transport>要素や<spam>要素の記述が参考となる。

プロセッサを経てのメール処理の流れは, 例えば

◆root プロセッサでスパムメールの条件に合致したメールは spam プロセッサへ

◆root プロセッサで goodboy@good-domain などの White List に合致する送信者からのメールは transport プロセッサへ

などとなる。メール中継などの設定は, 主として transport プロセッサブロックにメール処理の記述を追加・編集することから始まる。

●Matcher と Maillet のコンビネーション・パイプライン

各プロセッサ内でのメール処理は, Matcher と Maillet によるコンビネーション・パイプライン (Maillet パイプライン) の記述による。ここで, Matcher はメールを「ある条件にマッチする」, 或は「ある条件にマッチしない」といった風に分類するプログラムで, Maillet は分類されたメールを対象として具体的に処理するプログラムである。この Matcher と Maillet の組み合わせによりメール処理の流れを記述するのである。

Apache James には, All (全てのメールにマッチ) をはじめとする 26 の Matcher が, また, AddFooter (メールメッセージにテキストをフッターとして追加する) はじめ 19 の Maillet が予め組み込み提供されている (数字はドキュメントに具体的に記述されている Matcher,

Maillet の数を示す)。Matcher と Maillet プログラムを自作して Apache James に組み込み、機能を更に強化することも勿論可能であり、また、このようなことが可能なところが、Apache James の大きな特徴でもある。予め提供されている Matcher と Maillet の詳細については、ドキュメントを参照してほしい。

●Maillet パイプライン記述の具体例

プロセッサ内での Maillet パイプライン記述の概略を、config.xml ファイルの記述を参考に具体例を通して眺めてみよう。

迷惑メールはごみ箱へ

badboy@badhost.baddomain からの迷惑メールは、ごみ箱へ捨てて去ってしまうことにする。例えば、次のように書くことになるだろう。

```
<processor name="root">
===== 中略 =====
  <maillet match="SenderIs=badboy@badhost.baddomain" class="ToProcessor">
    <processor> spam </processor> ← spam プロセッサへ
    < notice>550 Requested action not taken: rejected </notice >
  </maillet>
===== 以下 略 =====
</processor>
<processor name="spam">
  <maillet match="All" class="Null"/> ← ごみ箱へ
</processor>
```

上の例では、maillet タグ内での記述方法や処理の流れが分かるように故意に冗長に記述している。SenderIs (送信者名がマッチ) と All (全てにマッチ) は Matcher プログラム名、ToProcessor (別のプロセッサへ処理を移す) と Null (何もしない) は Maillet プログラム名である。上の記述は、簡単に root プロセッサブロック内で次のように書けばすっきりする。

```
<processor name="root">
===== 中略 =====
  <maillet match="SenderIs=badboy@badhost.baddomain" class="Null"/>
===== 以下 略 =====
</processor>
```

メールの中継拒否

上述したように、デフォルト設定ではメールの中継は拒否された。これは、config.xml ファイルでの以下の記述によっている。

```
<processor name="root">
===== 中略 =====
<!-- Send remaining mails to the transport processor for either local or remote delivery -->
  <maillet match="All" class="ToProcessor"> ← 残りのメールは全て transport へ
    <processor> transport </processor>
  </maillet>
</processor>
<processor name="transport">
===== 中略 =====
  <maillet match="RemoteAddrNotInNetwork=127.0.0.1" class="ToProcessor">
    <processor> relay-denied </processor> ← ローカルホスト以外からは・・・
    <notice>550 - Requested action not taken: relaying denied</notice>
```

```
</mailet>
===== 以下 略 =====
</processor>
```

root プロセッサ内でスパムメールなどと判定されなかったメールは全て transport プロセッサに処理を移されるが、接続してきた SMTP クライアントのアドレスが 127.0.0.1 (ローカルホスト) 以外である場合は relay-denied プロセッサに処理を移され、結果として他サーバへの中継は拒否される、というものである。

■Apache James でのメール中継の設定

初期設定で禁止されているメールの中継について、幾つかの設定とその検証を試みる。設定は、全て config.xml ファイルで行う。

◆transport プロセッサブロック内での処理の流れ

先ず以下での説明の都合上、config.xml のデフォルト設定における transport プロセッサブロック内での Mailet パイプラインの流れを大雑把に見ておこう。次の 4 段階の構成となっている。

(1)<mailet match="RecipientIsLocal" class="LocalDelivery"/>

受信者がローカルユーザの場合は Mailet プログラム LocalDelivery へ。

(2)<mailet match="HostIsLocal" class="ToProcessor">

```
<processor> local-address-error </processor>
```

```
<notice>550 - Requested action not taken: no such user here</notice>
```

```
</mailet>
```

上記(1)でローカル配信されなかったメールの中、ローカルホスト宛のメールは user-unknown として local-address-error プロセッサへ。

(3)<mailet match="RemoteAddrNotInNetwork=127.0.0.1" class="ToProcessor">

```
<processor> relay-denied </processor>
```

```
<notice>550 - Requested action not taken: relaying denied</notice>
```

```
</mailet>
```

Matcher である RemoteAddrNotInNetwork の値として設定されていないネットワークアドレスからのメールは、メール中継拒否で relay-denied プロセッサへ。

(4)<mailet match="All" class="RemoteDelivery">

残りのメールは全て Mailet プログラム RemoteDelivery へ。

◆組織内 SMTP クライアントからの中継許可

組織内の SMTP クライアントからのメールは中継するよう設定する。上述したメール処理の流れから容易に分かるように、初期設定での transport プロセッサブロック内の記述を次のように書き換える (該当部分のみを抜粋) (*1, 重要)。

```
<mailet
match="RemoteAddrNotInNetwork=127.0.0.1, 192.168.0.*" class="ToProcessor">
  <processor> relay-denied </processor>
  <notice>550 - Requested action not taken: relaying denied</notice>
```

</mailet>

この設定で、組織内 (IP アドレス 192.168.0.*) SMTP クライアントからのメールは、relay-denied プロセッサに処理を移されない。

●テスト環境内のホストからテスト環境内ホストへの中継

図 2 に示したメールの中継テストを行う。メール受信者のアドレスを、r-user01@sl-epson.aoki-home.kagawa-u.ac.jp ではなく、r-user01@[192.168.0.8]とした^(*)2) ^(*)3)。メールは期待通りに中継された (以下参照)。

```
Return-Path: <s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp>
Received: from ravie01.aoki-home.kagawa-u.ac.jp ([192.168.0.4])
    by sl-epson.aoki-home.kagawa-u.ac.jp (JAMES SMTP Server 2.2.0) with
SMTP ID 137
    for <r-user01@[192.168.0.8]>;
    Tue, 2 Nov 2004 09:52:20 +0900 (JST)
Received: from end-pro-1000.aoki-home.kagawa-u.ac.jp ([192.168.0.2])
    by ravie01 (JAMES SMTP Server 2.2.0) with SMTP ID 800
    for <r-user01@[192.168.0.8]>;
    Tue, 2 Nov 2004 09:36:28 +0900 (JST)
===== 中略 =====
Delivered-To: r-user01@[192.168.0.8]
```

中継テスト 3 5, relay-35-mail, 2004.11.02

●テスト環境内のホストから外部ホストへの中継

テスト環境内のホストから、内部の ravie01 ホストの Apache James を中継 SMTP サーバとして外部サーバ宛にメールを送信する (図 4 参照)。

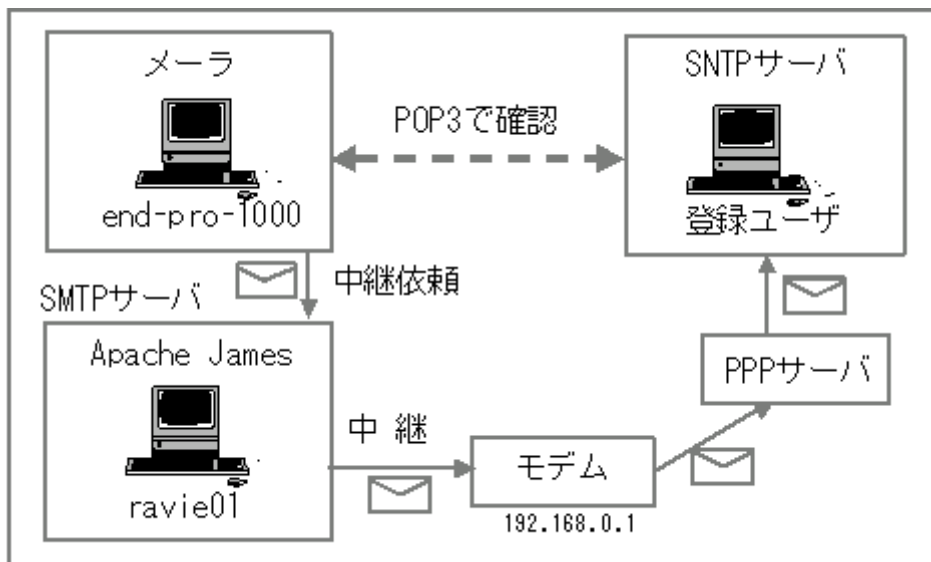


図 4 : テスト環境内から外部ホスト宛メールの中継

テストでは、図 4 での PPP サーバには本学総合情報基盤センター幸町分室内の PPP 接続サーバを、外部の SMTP サーバには筆者がアカウントを有する 3 つのプロバイダのサーバを選び、それぞれのサーバ内の筆者宛アカウントにメールを送信した。この場合、ravie01 ホストはモデムを通して DNS サーバのアドレスを知ることが出来るので、MX レコードとしての送信先 SMTP サーバのアドレスを得ることが可能となっている。

◆送信者名を s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp とした場合：

3つのプロバイダのうち、2つのプロバイダでメールの受信が断られた(*4)。受信を拒否されたメールについては、メールが受信されなかった旨を送信者宛に知らせる Postmaster からのメールが、送信者宛に送信されないまま ravie01 ホスト内のフォルダ内に残されている（この理由については、下記の注(*2)を参照のこと）。内容は次のようになっている（一部を抜粋）。

プロバイダ A 宛メールの場合 501 <s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp>... Sender domain must exist プロバイダ B 宛メールの場合 553 5.1.8 <s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp>... Domain of sender address s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp does not exist
--

上の例からも分かるように、メール受信にあたってエンベロープ送信者メールアドレスのドメイン部分がチェックされ、そのドメインが存在しない場合はメール受信を拒否している。返信先の確認とメールアドレスを詐称した迷惑メールの防止対策であろう。ただし、送信した SMTP サーバ（ここでは ravie01）そのものはチェック対象にはなっていない。ここでの中継拒否は、SMTP セッションレベルでのものである。

◆送信者名を aoki@xxx.kagawa-u.ac.jp とした場合：

3つのプロバイダ全てでメールは受信された。xxx.kagawa-u.ac.jp が実在するドメイン名だからである。送信側の ravie01.aoki-home.kagawa-u.ac.jp ホストが DNS サーバに登録されていないことは問題となっていない。この点では、個人的に立ち上げた SMTP サーバを中継してのメールで、送信者アドレスが詐称される余地がまだ残されているとも云えよう。いずれにせよ、テスト環境外へのメール中継機能が確認された

●メールゲートウェイの設定

図4では、ravie01 ホストは直接受信者の SMTP サーバに接続した。ravie01 から外部へのメール送信にあたって、一旦メールゲートウェイサーバを経由するよう設定する。Sendmail における SMART HOST の設定に相当する(文献5参照)。config.xml ファイルでは、Maillet プログラム RemoteDelivery のオプションとして次の個所に設定すればよい。

<pre> <maillet match="All" class="RemoteDelivery"> ===== 中 略 ===== <gateway> otherserver.mydomain.com </gateway> ← ここを書き換える <gatewayPort>25</gatewayPort> </maillet> </pre>

テストとして、本学のウィルスチェックサーバをメールゲートウェイに設定し、大学内の筆者のメールアドレス宛にメールを送信した。メールは、end-pro-1000 ホストのメーラ → ravie01 ホスト → ウィルスチェックサーバ → 筆者のアカウントがある学内サーバと期待通りの経路で受信された。

(*1) 「match="RemoteAddrNotInNetwork=127.0.0.1, 192.168.0.*"」のように、192.168.0.*を加えることで中継は許可されると考えたが、初期設定では思うように中継は許可されなかった。理由は、relay-denied プロセスが2重にチェックされる仕組みにある。

デフォルトでは config.xml ファイル中の SMTP server ブロック内で


```
<authorizedAddresses>127.0.0.0/8</authorizedAddresses>
```

となっており、このタグが有効である限り、ここに 192.168.0.0/24 の記述を追加しなければメールは中継されない。設定についてまとめておく。

◆<authorizedAddresses>要素が有効な場合：

<authorizedAddresses>要素内での設定が SMTP コネクションレベルのものであり優先されるので、メールの中継を設定するのであれば、この要素内にネットワークアドレス 192.168.0.0/24 等の記述を次のように必ず追加する。

```
<authorizedAddresses>127.0.0.0/8, 192.168.0.0/24</authorizedAddresses>
```

この場合には、

```
<mailet
  match="RemoteAddrNotInNetwork=127.0.0.1, 192.168.0.*" class="ToProcessor">
  <processor> relay-denied </processor>
  <notice>550 - Requested action not taken: relaying denied</notice>
</mailet>
```

の記述について次の2つの選択肢がある。

(1)全体をコメントとして無効にする。

(2)記述を残す。この場合には、ネットワークアドレス 192.168.0.*の記述を必ず追加する。アドレス 192.168.0.*を書き加えない場合にはメールは中継されない。このメール中継が許可されない場合でも、送信したメールは「メールを送信しました」と表示する。メールは確かにサーバに受け入れられており、送信者にはメールは中継されたように思えるのであるが、実はこのとき、メールはサーバの%JAMES_HOME%/apps/james/var/mail/relay-denied/に保存され（デフォルトの設定では保存するように設定されているが、例えば、保存せずに廃棄するよう設定も可能）、実際には中継されていない。

```
<notice>550 - Requested action not taken: relaying denied</notice>
```

の記述がどのように反映されるかは、relay-denied プロセッサブロックでの取り扱いによる。例えば、送信者宛に「中継できません」というメールを返す場合などには、そのときのエラーメッセージとして反映されるが、フォルダに保存したり、或いは廃棄する場合にはこの記述は意味をなさない。

◆<authorizedAddresses>要素がコメント扱いとして無効な場合：

transport ブロック内の「<mailet . . . class="ToProcessor">」の記述が有効となるので、メール中継を許可するのであれば、

```
match="RemoteAddrNotInNetwork=127.0.0.1, 192.168.0.*" class="ToProcessor">
```

のように、192.168.0.*を必ず追加記述する。

192.168.0.*の記述がなければ当然メールの中継は許されないが、この場合、送信者にはメールの中継が許されたように見えて、実は中継されずにメールがサーバ内に relay-denied メールとして保存されるのは上述の場合と同様である。

以上から、relay-denied プロセスに関しては<authorizedAddresses>要素の有効/無効のどちらか一方を選択することが望ましい。

(*2)通常ならばメールの宛先を r-user01@sl-epson.aoki-home.kagawa-u.ac.jp とすべきであるが、この宛先では James サーバで一旦中継メールとして受け入れられるものの sl-epson ホストへは転送されない。理由は ravie01 ホストでの DNS サーバに関わる設定にある。上述したように、テスト環境下には真の DNS サーバは設置されておらず、ravie01 ホストで DNS サーバとして設定されているのは、対外接続のためのモデム (192.168.0.1) となっている。このため、MX レコードとしての sl-epson.aoki-home.kagawa-u.ac.jp のアドレスが得られず、どのホストにメールを送ってよいのか解決できないのである。Maillet プログラム RemoteDelivery はあくまでも MX レコードを検索するようで、hosts ファイルへの記述では不十分である。メールはサーバ内の outgoing フォルダに保存され、期限がくるまで時折送信を試みることとなる。

(*3)FreeBSD がインストールされている sl-epson ホストならば、名前解決の順序が hosts ファイル、DNS サーバ (127.0.0.1, 192.168.0.1 (モデム)) となっているので、まず hosts ファイルを参照するのではないかと考え、sl-epson ホストを経由しての ravie-01 ホストへのメール中継テストを行った。結果は、やはり MX レコードとしての ravie01 ホストアドレスが得られないとの理由で実際にメールは中継されなかった。dnsserver についてのログの一部を示す。

```
02/11/04 11:05:36 INFO dnsserver: Couldn't resolve MX records for domain ravie0
1.aoki-home.kagawa-u.ac.jp.
02/11/04 11:06:16 ERROR dnsserver: Couldn't resolve IP address for host ravie01.
```

また、Maillet についてのログには次のような記録が残されている。

```
02/11/04 10:20:16 INFO James.Maillet: RemoteDelivery: Storing message Mail109935
7215829-2-to-ravie01.aoki-home.kagawa-u.ac.jp into outgoing after 2 retries
```

```

02/11/04 11:06:16 INFO James.Maillet: RemoteDelivery: No mail server found for:
ravie01.aoki-home.kagawa-u.ac.jp
02/11/04 11:06:16 INFO James.Maillet: RemoteDelivery: Temporary exception delive
ring mail (Mail1099357215829-2-to-ravie01.aoki-home.kagawa-u.ac.jp: javax.mail.M
essagingException: There are no DNS entries for the hostname ravie01.aoki-home.k
agawa-u.ac.jp. I cannot determine where to send this message.
    at org.apache.james.transport.maillets.RemoteDelivery.deliver(RemoteDeliv
ery.java:398)
    at org.apache.james.transport.maillets.RemoteDelivery.run(RemoteDelivery.
java:912)
    at java.lang.Thread.run(Thread.java:536)
02/11/04 11:06:16 INFO James.Maillet: RemoteDelivery: Storing message Mail109935

```

(*4)メールを受信したプロバイダでは、1日あたり数10通の迷惑メールが届く。その理由がなんとなく理解できたように思える。

◆組織外からの組織宛メールの中継許可

組織外からのメールであっても、組織内へのメールであれば中継を許可するよう設定する。この設定は、受信専用のサーバであれば不要であるが、例えば本学におけるウィルスチェックサーバのように、組織内へのメールゲートウェイサーバの役割を担う場合には必要である(図5参照)。

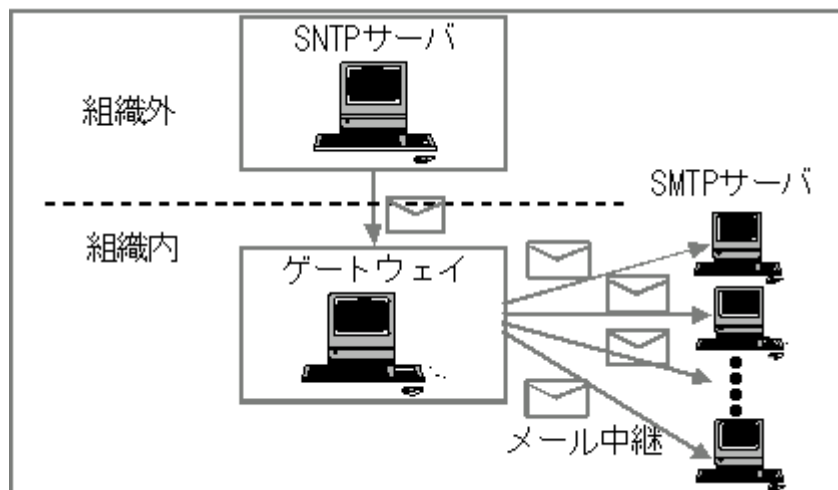


図5：組織外からの組織宛メールの中継

受信したメール取り扱いの流れを次のように設定する。

◆transport プロセッサ内で **Matcher** である **RemoteAddrNotInNetwork** を用いて中継を拒否されたメールの扱いを、初期設定のように **relay-denied** プロセッサではなく、新たに作成した **relay-allow-check** プロセッサとする。

◆**relay-allow-check** プロセッサブロックでは、メールが組織内のアドレス宛、或は中継を許す特定のアドレス宛かどうかを判定する。該当するメールは中継し、該当しないメールは、**relay-denied** プロセッサに処理を移す。この判定には、**Matcher** プログラム **HostIs**^(*)を利用する。

組織内への中継テスト1

テスト環境下で、ravie01 ホスト(192.168.0.4)で次のように設定してテストを行う。

◆自ホスト内ユーザ宛のメール及び中継を許可するホストからのメールを除く全てのメール

は、一旦 relay-allow-check プロセッサへ

◆relay-allow-check プロセッサ内では、end-pro-1000 ホスト (192.168.0.2) と sl-spson (192.168.0.8) 宛メールは Maillet プログラム RemoteDelivery へ、それ以外は relay-denied プロセッサへ(下記参照, 必要箇所のみ抜粋)。

```
transport プロセッサ内では
===== 中 略 =====
  <maillet match="RemoteAddrNotInNetwork=127.0.0.1" class="ToProcessor">
    <!--      maillet match="RemoteAddrNotInNetwork=127.0.0.1, 192.168.0.*"
class="ToProcessor" -->      ← 以下のテストでは、この部分はコメントとする
    <processor> relay-allow-check </processor>      ← このように変更する
    <notice>550 - Requested action not taken: relaying denied</notice>
  </maillet>
===== 以下 略 =====
relay-allow-check プロセッサ内では
<!-- My relay-allow-check processor , 2004.11.07 -->
<processor name="relay-allow-check">
===== 中 略 =====
<maillet match="HostIs=192.168.0.2, 192.168.0.8" class="RemoteDelivery">
  <outgoing> file://var/mail/outgoing/ </outgoing>
===== 中 略 =====
  </maillet>
< maillet match="All" class="ToProcessor">
  <processor> relay-denied </processor>
  <notice>550 - Requested action not taken: relaying denied</notice>
</maillet>
```

end-pro-1000 ホスト上のメーラを用い、SMTP サーバに ravie01 ホストを指定して、r-user01@[192.168.0.2]と r-user01@[192.168.0.8]宛にメールを送信する。メールは宛先に従い、それぞれのホストの r-user01 宛に中継された (以下参照)。

```
Return-Path: <s-user01@[192.168.0.2]>
Received: from ravie01.aoki-home.kagawa-u.ac.jp ([192.168.0.4])
  by sl-epson.aoki-home.kagawa-u.ac.jp (JAMES SMTP Server 2.2.0) with
SMTP ID 617
  for <r-user01@[192.168.0.8]>;
  Sun, 7 Nov 2004 13:44:53 +0900 (JST)
Received: from end-pro-1000.aoki-home.kagawa-u.ac.jp ([192.168.0.2])
  by ravie01 (JAMES SMTP Server 2.2.0) with SMTP ID 624;
  Sun, 7 Nov 2004 13:28:52 +0900 (JST)
Date: Sun, 07 Nov 2004 13:24:04 +0900
From: S-USER01 <s-user01@[192.168.0.2]>
===== 中 略 =====
To: r-user01@[192.168.0.2], r-user01@[192.168.0.8]
===== 中 略 =====
Delivered-To: r-user01@[192.168.0.8]
===== 中 略 =====
中継テスト66
relay-test-66
2004.11.07
```

組織内への中継テスト2

許可する中継先ホストに、本学情報基盤センター幸町分室の xxx.kagawa-u.ac.jp を付け加える。このためには、relay-allow-check プロセッサ内に次のような<maillet>要素を書き加えるとよい(*2) (以下参照)。

```

<mailet match="HostIs=xxx.kagawa-u.ac.jp" class="RemoteDelivery">
  <outgoing> file://var/mail/outgoing/ </outgoing>
  ===== 中 略 =====
  <gateway> gateway-server.kagawa-u.ac.jp </gateway> ← ここには実際のホスト名
  を書く
  <gatewayPort>25</gatewayPort>
</mailet>

```

「組織内への中継テスト1」と同様に、end-pro-1000 ホスト上のメーラを用い ravie01 ホストを SMTP サーバに指定して、宛先を aoki@xxx.kagawa-u.ac.jp、送信者名を s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp としてメールを送信した。メールは xxx.kagawa-u.ac.jp ホストで受信、aoki 宛に配信された。

(*1)Matcher プログラム HostIs は、メールの宛先（送信先）のホスト名が一致するかどうかを判定するものである。HostIs=に与える host 名（のリスト）には、例えば 192.168.0.*や*.kagawa-u.ac.jp 風のワイルドカード表記は許されない。1つ1つのホスト名をカンマか空白で区切って記述する必要がある。

(*2)xxx.kagawa-u.ac.jp 宛にメールを送信するには、メールゲートウェイの役目をするウィルスチェックサーバを経由する必要がある。

■エラーメール等の取り扱い

上述の「◆Spool Manager ブロックでのメール取り扱いの流れ」でも説明したように、受信されたメールは root, transport プロセッサで判別された後自ホスト内のユーザに配信されるか他のサーバに中継・送信されるか、或いは他のプロセッサに処理を移される。ここでは、各プロセッサに移されたエラーメール等の取り扱いについて説明する。

◆初期設定におけるプロセッサブロック内での処理の流れ

各プロセッサに送られたメールが初期設定ではどのように処理されているのかを、local-address-error プロセッサを例にとり見てみることにする（以下参照、必要部分のみ抜粋）。このプロセッサは、自ホスト宛に送られてきたメールのうち、受信者が不在ということで user-unknown として処理されるメールを取り扱う。

```

<!-- This processor handles messages that are for local domains, where the user is
unknown -->
  <processor name="local-address-error">
    <!--
      <mailet match="All" class="Bounce">          ← (1)
        <attachment>none</attachment>
      </mailet>
    -->
    <!--
      <mailet match="All" class="NotifyPostmaster"/> ← (2)
    -->
    <mailet match="All" class="ToRepository">
      <repositoryPath> file://var/mail/address-error/</repositoryPath> ← (3)
    <!--
      <repositoryPath> db://maildb/deadletter/address-error </repositoryPath>
    -->          ← (4)
    </mailet>
  </processor>

```

初期設定でコメント扱いとされていないのは、(3)の「全てのメールは address-error ディレクトリへ保存する」の設定のみである。この設定により、user unknown メールは全て保

存されることとなる。

殆どのプロセッサで同じ記述がなされているので、(1)、(2)、(4)について説明しておく。(1)は、送信者に user unknown としてエラーとなった旨知らせるメールを作成する設定である。<attachment>none</attachment>は、メール本文を添付しないことを示している。このタグをコメントとすれば、本文が添付される(*1)。

(2)は、postmaster にエラーを知らせるメールを作成し送信する設定である。この設定を用いる場合には、予め postmaster を（そして、その alias をも）ユーザ登録していないと、当然のことではあるが宛先不明でエラーとなる。

(4)は、メールをディレクトリ内のファイルとしてではなく、データベース(RDB)に保存する際の設定である。

以上から分かるように、初期設定のままではエラーメールはフォルダ内に保存される。しかしながら、全てのエラーメールをこのように保存する設定は、ログとして残すという意味では有用ではあるが、実際の運用では到底適切とは云えないであろう。

(*1) (1)の <mailet match="All" class="Bounce"> <attachment>none</attachment> に代えて、<mailet match="All" class="NotifySender"/>と記述しても同様の結果が得られる。ただし、この場合には、メール本文が添付される。

◆各プロセッサブロック内での設定例

各プロセッサ内でメールをどのように処理すればよいかを例示する。あくまでも例示であり、実際の運用状況により設定を初期設定のままとするなり、或いは目的に応じて適宜変更してほしい。

●error プロセッサ

このプロセッサ内ではエラーとなったメールが処理されるが、エラーが Mailet パイプラインの問題に起因する場合だけでなく、システムの内部的な問題による場合もあり得ることを考慮して、少なくとも Postmaster 宛にはエラーが生じた旨のメッセージを送信することとする。config.xml ファイルの初期設定におけるコメントを全てアンコメントとしてみる(以下参照) (*1)。

```
<processor name="error">
  <mailet match="All" class="Bounce"/>
  <mailet match="All" class="NotifyPostmaster"/>
  <mailet match="All" class="ToRepository">
    <repositoryPath> file://var/mail/error/</repositoryPath>
  </mailet>
</processor>
```

(*1)transport プロセッサブロック内で、

```
<mailet match="RemoteAddrNotInNetwork=127.0.0.1" class="ToProcessor">
```

の記述が有効となっている場合には、アドレス 127.0.0.1 に加えて、自ホストの IP アドレスをも、例えば"---=127.0.0.1, 192.168.0.4"のように、記述しておく。自ホスト外のユーザへの”Bounce”の場合、この記述がないと送信に失敗する。

●spam プロセッサ

spam メールであるかどうかの判定は簡単ではないが、真に spam メールを判定されたメールは保存するのではなく、全てを“ごみ箱”に捨て去って構わないであろう。この場合、単純に次のように設定する。

```
<processor name="spam">
  <mailet match="All" class="Null"/>
</processor>
```

●relay-denied プロセッサ

上述の「◆組織外からの組織内宛メールの中継許可」の項で、中継を拒否されたメールを直接 relay-denied プロセッサに処理を移すのではなく、新たに作成した relay-allow-check プロセッサに迂回させ、その後不要なメールを relay-denied プロセッサで処理する方法を紹介した。この方法を採用するならば、relay-denied プロセッサでは殊更何かの処理を行う必要は無いものと思われる。また、SMTP サーバ設定での<authorizedAddresses>タグを用いて SMTP コネクションレベルでのメール中継拒否を取り扱うことができるので (Ver 2.2.x 以降)、relay-denied プロセッサに大きな処理の役目を課す必要もないであろう。メールを全て消去してしまうこととする (下記参照)。

```
<processor name="relay-denied">
  <mailet match="All" class="Null"/>
</processor>
```

●local-address-error プロセッサ

user unknown となったメールを取り扱うブロックであり、全てのメールを“ごみ箱”に捨て去ってもよいであろうが、それでは余りにも能が無さ過ぎる。次のような設定を考えてみる。

- (1)接続してきた SMTP クライアントが組織外 (例えば 192.168.0.*以外とする) のホストである場合には、エラーメールがスパムメールである可能性もあるので、全てを廃棄する。
- (2)送信者名が“空”であるメールは廃棄する(*1)。
- (3)送信者名が組織内ユーザのものであるメールには、エラーとなった旨の簡単なメッセージを送信する (ここでは、該当する送信者名が***@***.aoki-home.kagawa-u.ac.jp であるとする) (*2)。
- (4)Postmaster には通知しない。
- (5)もし残りのメールがあれば、それらは“ごみ箱”へと廃棄する。

設定内容は次の通りである (下記参照)。下記リスト中に説明のため付した数字(1)~(5)は、上記の設定内容を示す数字(1)~(5)にそれぞれ対応している。

```
<processor name="local-address-error">
<mailet match="RemoteAddrNotInNetwork=127.0.0.1, 192.168.0.*" class="Null"> (1)
  <mailet match="SenderIsNull" class="Null"> (2)
  <!-- The original message is not attached to keep the bounce processor from
delivering spam -->
    <mailet match="SenderIsRegex=.*@.*¥.aoki-home¥.kagawa-u¥.ac¥.jp"
class="Bounce"> (3)
      <attachment>none</attachment>
    </mailet>
  <!-- mailet match="All" class="NotifyPostmaster"/ --> (4)
  <mailet match="All" class="Null"/> (5)
</processor>
```

(*1)ここで使われている“SenderIsNull”はドキュメントには記載されていないが、Ver.2.2から提供されている Matcher で (<http://james.apache.org/download.cgi>での「New Feature」を参照)、送信者名が“空”であるメールにマッチする。

(*2) “SenderIsRegex” もドキュメントには記載されていないが、Ver.2.2 から提供の **Matcher** である (上記と同様「New Feature」を参照)。マッチすべき送信者名の記述に正規表現を用いることができる。

■信頼できない送信者アドレスメールの取り扱い

ここでは、メール取り扱いの出発点となる **root** プロセッサ内の記述と、そこでスパムメール排除の目的で用いられている、**Matcher** プログラム **SenderInFakeDomain** について紹介する。また、「組織内から偽りの送信者名をもつメールの送信を許さない」設定についてのヒントも併せて紹介する。

◆初期設定における **root** プロセッサ内での処理の流れ

root プロセッサでのメール処理の流れは、初期設定では次のように構成されている。◆送信者のメールアドレスのホスト名が信頼できるドメインに属しているかどうかを **Matcher** プログラム **SenderInFakeDomain** を用いてチェックする。スパムメールを排除するのに有効である。記述された偽りのドメインアドレスのリストに合致するメールは、直ちに **spam** プロセッサに処理を移す。コメント扱いとなっている。

◆メール中継がループに陥らないよう中継回数を制限し、制限を越えたメールは廃棄する。初期設定では、中継回数の制限値は 30 回である。

◆全てのメールを **Maillet** プログラム **XMLVirtualUserTable** に渡し、データベース情報に従ってメールアドレスをエイリアス変換する(*1)。コメント扱いとなっている。

◆**White List** に含まれるなどの、明らかに信頼できる送信者からのメールは、**transport** プロセッサに処理を移す。コメント扱いとなっている。

◆**Black List** に含まれる送信者からのメールは **spam** プロセッサへ。コメント扱いとなっている。

◆残りのメールは全て **transport** プロセッサに処理を移す。

(*1) **Maillet** プログラム **XMLVirtualUserTable** については、Ver.2.1 のドキュメント http://james.apache.org/provided_matchers_2_1.html には記載が無く、よく似た **Maillet** として **JDBCVirtualUserTable** が記載されている。

◆**Matcher** プログラム **SenderInFakeDomain**

root プロセッサ内の冒頭でスパムメール排除の目的で用いられる **Matcher** プログラム **SenderInFakeDomain** の振る舞いについて検証する。

●**config.xml** における初期設定

この **Matcher** は、送信者のメールアドレスのホスト名が信頼できるドメインに属しているかどうかをチェックする。初期設定ではコメント扱いとなっているが、次のように記述されている。

```
<maillet
match="SenderInFakeDomain=64.55.105.9,64.94.110.11,194.205.62.122, . . .
, 203.119.4.6" class="ToProcessor">
  <processor> spam </processor>
</maillet>
```

この記述からも分かるように、**SenderInFakeDomain** のパラメータとしては IP アドレスの

リストが用いられている。この IP アドレスのリストのいずれかに送信者メールアドレスのホスト名から得られる IP アドレスが一致する場合、そうしたメールは spam プロセッサに処理を移される設定となっているのである。記述の仕方も明瞭である。

このように、SenderInFakeDomain は信頼できる送信者であるかどうかを判断するに便利な Matcher であるが、信頼できない IP アドレスの全てをリストアップするのは実際には容易でない。また、送信者アドレスをチェックするといっても、どのようにチェックされるのか、チェックに用いられる送信者メールアドレスとはどのようなものなのか等について、もう少し詳しく検証する必要がある。

●基本的な振る舞い

SenderInFakeDomain の振る舞いを調べるについて参考となるのは、Ver.2.1 についてのドキュメントである^(*) (Ver.2.2 のドキュメントはアップロードされていない)。次のように紹介されている。

SenderInFakeDomain

Description: Matches mails where the host name in the address of the sender cannot be resolved. All recipients are returned.

Configuration string: None.

ここでは、パラメータが *None* となっていることに注目し、幾つかの検証のためのテストを行ってみることとする。以下では、SMTP サーバの root プロセッサ内での設定を

```
<mailet match="SenderInFakeDomain" class="ToProcessor">
  <processor> spam </processor>
</mailet>
```

とし、spam プロセッサでは全てのメールを spam フォルダに保存する設定とする。

テスト1 - 送信者アドレス s-user01@[192.168.0.2] -

end-pro-1000 ホスト (192.168.0.2) 上のメーラを用い、SMTP サーバとして revie01 ホストを指定して、ravie01 ホストのユーザ r-user01 宛に直接メールを送信する。

エンベロープ送信者名: s-user01@[192.168.0.2]

ヘッダの「From:」項: s-user01@[192.168.0.2]

である。メールは spam フォルダに移されることなく r-user01 宛に配信された。

テスト2 - 送信者アドレス s-user01@endpro-1000.aoki-home.kagawa-u.ac.jp -

上記テスト1と同様に、ravie01 ホストのユーザ r-user01 宛に直接メールを送信する。

エンベロープ送信者名: s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp

ヘッダの「From:」項: s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp

である。メールは SenderInFakeDomain のチェックにより spam フォルダに移された。このときの mailet に関連するログの内容を示す (下記参照)。

```
11/11/04 10:48:52 INFO James.Maillet: SenderInFakeDomain: No MX, A, or CNAME
record found for domain: end-pro-1000.aoki-home.kagawa-u.ac.jp
11/11/04 10:48:52 INFO James.Maillet: ToRepository: Storing mail
Mail1100137702015-2 in file://var/mail/spam/
```

これから、送信者のメールアドレスのホスト名

end-pro-1000.aoki-home.kagawa-u.ac.jp

が DNS サーバへの照会によりチェックされていることが分かる。また、DNS サーバへの問い合わせは、MX, A, CNAME レコードについてであることも分かる。

テスト3 — (エンベロープ From) と (ヘッダ From :) —

上記の2つのテストで、送信者メールアドレスのホスト名が DNS サーバに登録されているかどうかによってチェックされていることが判明した。それでは、エンベロープ送信者名 (SMTP コネクション時における From . . .) とメールヘッダの「From :」に記された送信者名のどちらがチェックされるのであろうか。エンベロープ送信者名によってメールが中継されていることはよく知られているところであり、従って SenderInFakeDomain によるチェック対象もエンベロープ送信者名であることが予想されるのであるが、ここではそのことを改めて確認する。

上記テスト1, 2と同様に, ravie01 ホストのユーザ r-user01 宛に直接メールを送信する。ただし、今回は

エンベロープ送信者名 : s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp

ヘッダの「From : 項」: s-user01@[192.168.0.2]

である。メールは SenderInFakeDomain のチェックにより spam フォルダに移された。このことから、エンベロープ送信者名がチェックの対象となっていることが再確認された^(*)2)。関連するログとメール内容を示す (下記参照)。

```

===== mailer ログから =====
12/11/04 06:36:12 INFO James.Maillet: SenderInFakeDomain: No MX, A, or CNAME
record found for domain: end-pro-1000.aoki-home.kagawa-u.ac.jp
12/11/04 06:36:12 INFO James.Maillet: ToRepository: Storing mail
Mail1100208965112-0 in file://var/mail/spam/
===== このとき送信したメール =====
Return-Path: s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp ← エンベロープ送
信者名
Received: from end-pro-1000.aoki-home.kagawa-u.ac.jp ([192.168.0.2])
        by ravie01 (JAMES SMTP Server 2.2.0) with SMTP ID 434
        for <r-user01@[192.168.0.4]>;
        Fri, 12 Nov 2004 06:36:05 +0900 (JST)
Date: Fri, 12 Nov 2004 06:29:31 +0900
From: S-USER01 s-user01@[192.168.0.2] ← ヘッダに記載された送信者名
===== 中略 =====
To: r-user01@[192.168.0.4]
Subject: FakeDomail Test 04
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit

FakeDomail Test 04
2004.11.12

```

テスト4 — 中継サーバには無関係 —

テスト3の結果から、エンベロープ送信者名が SenderInFakeDomain によるチェック対象であることが明らかになった。このことを再確認する意味で、メールを中継させ、中継するサーバには無関係であることを示しておこう。メールを
end-pro-1000 ホスト → sl-epson ホスト → ravie01 ホスト

へと中継させる。このときの結果は次のように期待されたものであった。

エンベロープ送信者名：s-user01@[192.168.0.2] → ravie01 で受信者に配信

エンベロープ送信者名：s-user01@end-pro-100.aoki-home.kagawa-u.ac.jp

→ (end-pro-1000.aoki-home.kagawa-u.ac.jp の MX, A, CNAME レコード不在)

→ ravie01 の spam フォルダへ

(*1) http://james.apache.org/provided_matchers_2_1.html

(*2) メール内容からも分かるように、エンベロープ送信者名とヘッダの「From:」項の送信者名を異なるようにすることは簡単である。ヘッダの記載内容を安易に信じてはならない。

(*3) 上の結果は、逆のケース、すなわち、

エンベロープ送信者名： s-user01@[192.168.0.2]

ヘッダの「From: 項」： s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp

としたメールは、SenderInFakeDomain のチェックを逃れることが可能であることを示している。多くの人は、ヘッダに記された「From: 送信者メールアドレス」を信じがちである。

エンベロープ送信者名： 偽りのユーザ名@実在のドメイン名

ヘッダの「From: 項」： 他人のメールアドレス、または偽りのメールアドレス

等の場合を想像してほしい。迷惑メールを排除することの難しさを示している。

●パラメータ設定時の振る舞い

SenderInFakeDomain について、初期設定にもあるように、パラメータとして IP アドレスのリストを設定した場合の振る舞いについて検証する。以下のテストでは簡単のため、SMTP サーバの root プロセッサ内での設定を

```
<mailet match="SenderInFakeDomain=192.168.0.2" class="ToProcessor">
  <processor> spam </processor>
</mailet>
```

とする (192.168.0.2 は end-pro-1000 ホストの IP アドレス)。spam プロセッサでは、全てのメールを spam フォルダに保存するよう設定している。テストでは、end-pro-1000 ホスト上のメールを用い、SMTP サーバには revie01 ホストを指定して、ravie01 ホストのユーザ r-user01 宛に直接メールを送信する。

テスト 1

エンベロープ送信者等を次の様に設定する。

エンベロープ送信者名： s-user01@[192.168.0.2]

ヘッダの「From: 項」： s-user01@[192.168.0.2]

メールは、エンベロープ送信者名のホストアドレスが 192.168.0.2 に一致するとして、spam フォルダに移された。このときの mailet 関連のログを次に示す。

```
13/11/04 06:09:44 INFO James.Mailet: SenderInFakeDomain: Banned IP found for
domain: 192.168.0.2
13/11/04 06:09:44 INFO James.Mailet: SenderInFakeDomain: --> :192.168.0.2
13/11/04 06:09:44 INFO James.Mailet: ToRepository: Storing mail
Mail1100293768550-0 in file://var/mail/spam/
```

テスト 2

送信者アドレスのホスト部が 192.168.0.2 以外である場合にどのように取り扱われるかを見るため、エンベロープ送信者名等を次の様に設定し、上記テスト 1 と同様に、ravie01 ホストのユーザ r-user01 宛に直接メールを送信する。

エンベロープ送信者名： s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp

ヘッダの「From: 項」: s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp
 メールは SenderInFakeDomain のチェックにより、ドメインが DNS レコードに見つからないとして spam フォルダに移された。このときのログの内容を示す (下記参照)。

```
13/11/04 06:20:43 INFO James.Maillet: SenderInFakeDomain: No MX, A, or CNAME
record found for domain: end-pro-1000.aoki-home.kagawa-u.ac.jp
13/11/04 06:20:43 INFO James.Maillet: ToRepository: Storing mail
Mail1100294413468-1 in file://var/mail/spam/
```

以上の2つのテストから、SenderInFakeDomain のパラメータとして IP アドレスのリストを設定した場合、エンベロープ送信者名のホストアドレス部が、

◆パラメータとして記された IP アドレスリストのどれかに一致するかどうか

◆DNS サーバへの問い合わせにより、登録されているかどうか

の2通りの方法でその信頼性をチェックされることが分かる。だとすれば、初期設定のコメントをはずして有効な設定とし、信頼できないエンベロープ送信者名を持つメールを排除する設定としても他に不都合を生じないであろう。「組織内から偽りの送信者名をもつメールの送信を許さない」設定としても有効である。勿論、組織外から届くメールについては、返信することも出来ないメールであることから、spam プロセッサ内で廃棄処理とすることで一向に差し支えない。

◆組織内からの送信者アドレス詐称メールの排除

SMTP サーバを管理するにあたっては、アドレスを詐称した組織外からのメールを受け取らないだけでなく、組織内から送信者アドレスを詐称したメールの組織外への送信を許さないことが強く求められる。この設定は、CF プログラムを用いた sendmail の設定での、CLIENT_HOST_*となっているものに相当する (文献5参照)。これまで述べてきた各種設定を参考として、こうした設定についてのヒントを以下に紹介する。送信者名については、環境に合わせて適当に読み替えてほしい。

●Matcher プログラム “SenderInFakeDomain” を利用する

上述したように、root プロセッサ内の最初で SenderInFakeDomain を用い、取り扱うメールの全てについてエンベロープ送信者アドレスのドメイン部が信頼できるかどうかをチェックする。

●Matcher プログラム “SenderHostIs” を利用する^(*)

組織外へのアドレス詐称メールの送信を排除することを目的として、transport プロセッサ内で用いる。用いるのは、外部へメールを送信する

```
<mailet match="All" class="RemoteDelivery">
  <outgoing> file://var/mail/outgoing/ </outgoing>
  ===== 中 略 =====
</mailet>
```

の個所がよいであろう。match="All"の部分

```
match="SenderHostIs=end-pro-1000.aoki-home.kagawa-u.ac.jp,
ravie01.aoki-home.kagawa-u.ac.jp, sl-epson.aoki-home.kagawa-u.ac.jp"
```

のように記述する^(*)。ただし、この Mailet パイプラインで扱う全てのメールが組織内のユ

ーザから組織外へと送信されるメールであることが前提であり、組織外からのメールを組織内の他の SMTP サーバへ中継する場合などは、このパイプラインを通過しない工夫が必要である（上述した「◆組織外からの組織内宛メールの中継許可」における relay-allow-check プロセッサの作成を参照）。

●Matcher プログラム “SenderIsRegex” を利用する

上記の “SenderHostIs” を “SenderIsRegex” で置き換える。

```
match="SenderIsRegex=.*@.*aoki-home¥.kagawa-u¥.ac¥.jp"
```

のように正規表現を用いて記述し、***@***.aoki-home.kagawa-u.ac.jp のエンベロープ送信者名メールのみを、外部へ送信する。

(*1) “SenderIsRegex” と同様、Ver.2.2 で提供された Matcher である。

(*2)送信者アドレスが組織内のものに合致せず送信されなかった残りのメールについての措置が必要である。例えば、<mailet match="All" class="Null"/>などを transport プロセッサの末尾に記述する。

■件名や添付ファイルによるメールのフィルタリング

メールの件名（題名，Subject）の内容や添付ファイルの有無によりメールを選別する方法等を簡単に紹介する。こうした機能は、提供されている Matcher プログラムを用いることで容易に実現される。

◆Subject によるフィルタリング

用いられる Matcher は、ヘッダの Subject 内容が与えられた文字列に一致するメールを選別する “SubjectIs” と、Subject の内容が与えられた文字列で始まるメールを選別する “SubjectStartWith” である。それぞれの利用法について説明する(*1)。

SubjectIs

```
<mailet match="SubjectIs=subject-string" class="class-name">
```

と記述する。例えば、

```
<mailet match="SubjectIs=this is a spam-mail" class="ToProcessor">
  <processor> spam </processor>
</mailet>
```

であれば、件名が “this is a spam-mail” であるメールは spam プロセッサに処理を移される。

SubjectStartsWith

```
<mailet match="SubjectStartsWith=subject-start-string" class="class-name">
```

と記述する。例えば、次のように用いると、“Software”で始まる件名のメールは spam プロセッサ送りとなる。

```
<mailet match="SubjectStartsWith=Software" class="ToProcessor">
  <processor> spam </processor>
</mailet>
```

◆Subject 内容が日本語の場合のフィルタリング

日本語で書かれた件名をチェックしようとする場合には、用いる文字エンコーディングを config.xml ファイル内で指定しなければならない。インストール時の config.xml では文字エ

ンコーディングが指定されていないので、チェック対象の文字列を UTF-8 エンコーディングで表さなければ Apache James 起動時にエラーを生じる。

以下では、Windows2000/XP 上での config.xml ファイル編集を例にとり、日本語 Subject によるフィルタリングについて説明する。Windows2000/XP では、よく知られているように、デフォルトで用いられるエンコーディングは Shift_JIS である。

●config.xml ファイルでの記述方法

まず、先頭行に属性 encoding の値を次のように設定する(*1)。

```
<?xml version="1.0" encoding="Windows-31J"?>
```

その上で、“SubjectIs”については、例えば

```
<mailet match="SubjectIs=これはスパムメール" class="ToProcessor">
  <processor> spam </processor>
</mailet>
```

のように記述し、“SubjectStartsWith”については

```
<mailet match="SubjectStartsWith=ソフトウェア" class="ToProcessor">
  <processor> spam </processor>
</mailet>
```

とする(*2)。

(*1)Windows で用いられる文字エンコーディングは Shift_JIS であるが、

```
<?xml version="1.0" encoding="Windows-31J"?>
```

のように設定する。Java プログラミングによる場合、encoding="Shift_JIS"の設定では、文字コードの違いから I, II, ~, ①, ②などの文字が文字化けする。

(*2)2つ以上の条件を設定する場合、文字列をリストとして記述することは許されていない。1つ1つの条件を個別に設定する必要がある。

●フィルタリングの結果例

上記のような設定下での結果について次に例を示す。以下で示す例は、一旦 spam フォルダに処理を移されたメールにヘッダとフッタを付加した後(後出)、再度受信者に配信するよう設定して得られたものである。

SubjectIs のテスト例

Subject に“これはスパムメール”を設定してユーザ r-user01 宛に送信した。メーラを用いて受信し表示した内容を示す。

```
Return-Path: <s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp>
Received: from end-pro-1000.aoki-home.kagawa-u.ac.jp ([192.168.0.2])
  by ravie01 (JAMES SMTP Server 2.2.0) with SMTP ID 919
  for <r-user01@[192.168.0.4]>;
  Tue, 16 Nov 2004 08:08:18 +0900 (JST)
Message-ID: <14806696.1100560098719.JavaMail.aoki@ravie01>
Date: Tue, 16 Nov 2004 08:03:14 +0900
From: S-USER01 <s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp>
==== 中 略 ====
To: r-user01@[192.168.0.4]
Subject: これはスパムメール ← Subject の内容が条件に一致
Content-Type: text/plain; charset=Shift_JIS
Content-Transfer-Encoding: quoted-printable
```

SPAM: MAY-BE-SPAM	← 付加されたヘッダ
Delivered-To: r-user01@[192.168.0.4]	
フィルタリングのテスト 1	
2004.11.16	
SPAM: May be Spam	← 付加されたフッタ

SubjectStartsWith のテスト例

Subject に “ソフトウェア, 全て 70%OFF” が設定されたメールの内容を示す。

Return-Path: <s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp>	
===== 中略 =====	
To: r-user01@[192.168.0.4]	
Subject: ソフトウェア, 全て 70%OFF	← Subject が “ソフトウェア” で始まっている
Content-Type: text/plain; charset=Shift_JIS	
Content-Transfer-Encoding: quoted-printable	
SPAM: MAY-BE-SPAM	← 付加されたヘッダ
Delivered-To: r-user01@[192.168.0.4]	
フィルタリングのテスト 2	
2004.11.16	
SPAM: May be Spam	← 付加されたフッタ

◆ヘッダの項目によるフィルタリング

用いられる Matcher は “HasHeader” で、特殊なヘッダ項目を有するメールを選別する。次のように用いる。

<pre><mailet match="HasHeader=Special-Header" class="ToProcessor"> <processor> spam </processor> </mailet></pre>
--

◆添付ファイルの有無や添付ファイル名によるフィルタリング

用いられる Matcher は, “HasAttachment” と “AttachmentFileNameIs” である。

HasAttachment

MIME type が “multipart/mixed” であるかどうかにより添付ファイルの有無をチェックし, メールを選別する。

```
<mailet match="HasAttachment" class="mailet-class-name">
```

と記述する。

AttachmentFileNameIs

添付されたファイルが特別なファイル名するときメールを拒否する場合などに便利な Matcher で, ファイル名として, 例えば *.src, *.bat, *.pif, *.exe などの様にワイルドカードを用いて記述できる。複数のファイル名を記述する場合には, カンマかスペースで区切ってリストとして,

```
<mailet match="AttachmentFileNameIs=*.exe, *.doc" class="class-name">
```

のように記述する。

◆フィルタリングされたメールの取り扱いーヘッダとフッタの附加ー

何らかの形でフィルタリングされ, 例えば迷惑メールと判定されたようなメールはどのように取り扱うとよいのだろうか。迷惑メールであるかどうかの正しい判定は困難で, 正しい

メールを間違って迷惑メールと判断する誤検知 (False Positive) が常に起こり得る。また、組織内から外部に出て行くメールであるか、外部から組織内に届くメールであるかによっても取り扱いは異なるであろう。何れにせよ、フィルタリングにより検知されたメールを spam プロセッサで単に破棄するだけでは十分ではない。

ここでは、外部から組織内のユーザに届いたメールにはヘッダあるいはフッタを附加してユーザに注意を喚起する場合を想定する。

●ヘッダの附加

Maillet プログラム “AddHeader” を用いて次の様に記述する。これで、ヘッダに SPAM: MAY-BE-SPAM 項が付け加えられる。

```
<mailet match="RecipientIsLocal" class="AddHeader">
  <name> SPAM </name>
  <value> MAY-BE-SPAM </value>
</mailet>
```

●フッタの附加

用いる Maillet プログラムは “AddFooter” である。例えば次の様に用いると、フッタとして “SPAM: May be Spam” が附加される。

```
<mailet match="RecipientIsLocal" class="AddFooter">
  <text> SPAM: May be Spam </text/>
</mailet>
```

件名 (Subject) が文字列 “Software” で始まるために検知され、ヘッダとフッタが附加されたメールの例を示す (以下参照)。

```
Return-Path: <s-user01@[192.168.0.2]>
Received: from end-pro-1000.aoki-home.kagawa-u.ac.jp ([192.168.0.2])
  by ravie01 (JAMES SMTP Server 2.2.0) with SMTP ID 948
  for <r-user01@[192.168.0.4]>;
  Sun, 14 Nov 2004 05:12:40 +0900 (JST)
Date: Sun, 14 Nov 2004 05:07:38 +0900
From: S-USER01 <s-user01@[192.168.0.2]>
===== 中 略 =====
To: r-user01@[192.168.0.4]
Subject: Software 70% off ← “Software” で始まる Subject
===== 中 略 =====
Message-ID: <6934571.1100376760657.JavaMail.aoki@ravie01>
SPAM: MAY-BE-SPAM ← 附加されたヘッダ

Footer-Header Test 01 ← 本文の始まり
2004.11.14
SPAM: May be Spam ← 附加されたフッタ
```

■おわりに

Apache James について、その SMTP サーバとしての基本設定と動作検証について紹介した。本稿での内容については、テスト環境内を中心としてではあるが、逐一検証した結果に基づいている。一読して頂ければ、Windows2000/XP, Linux, FreeBSD の各 OS 上で、利用環境にあった設定のもと、セキュアな SMTP サーバ構築が簡単にできるものと思っている。是非 Apache James を楽しんで頂きたい。

本稿では、Apache James のもつ拡張的な機能については余り触れていない。それらにつ

いてはあらためて別の機会に紹介したい。

参考文献

- (1)青木 昌三：「Java による Web アプリ開発の勧め（Ⅱ）－ Cocoon2 利用による Web ページ作成－」 香川大学総合情報基盤センター年報 第1号（2004年2月）
- (2)黒住 幸光：「Jakarta 活用指南 第16回 メールサーバ「Apache James を使う」」 JavaWorld 2004 July, アイ・ディ・ジー・ジャパン
- (3)菊田 英明：「SE・プログラマ スタートアップテキスト JSP [基礎]」 2002.2, 技術評論社（ISBN 4-7741-1340-9）
- (4)青木 昌三：「FreeBSD におけるメール関連プログラム－その設定と動作確認－」
青木 昌三：「qmail－設定と動作確認－」
香川大学情報処理センター 年報 第7号（1999年2月）
- (5)青木 昌三：「SMTP サーバプログラム Sendmail の設定(I)－
m4 マクロによる sendmail.cf ファイルの設定－」
青木 昌三：「SMTP サーバプログラム Sendmail の設定(IV)－
ウイルスチェックサーバへのメール転送－」
香川大学情報処理センター 年報 第11号（2003年2月）

Apache James の設定と動作検証 (II)

－拡張的機能の設定－

教育学部 (人間環境教育)

青木 昌三

■はじめに

別掲記事「Apache James の設定と動作検証 (I)－SMTP サーバとしての基本的な設定－」において、Apache Software Foundation(ASF) の James プロジェクトによる電子メールアプリケーションサーバ Apache James の基本的な設定と動作検証について紹介した⁽¹⁾。本稿では、Apache James のもつ SMTP サーバとしての拡張的な機能の設定とその動作検証を中心に紹介する。取り扱う内容は次の通りである。

- テスト環境の構成
- メーリングリストの設定
- データベース MySQL の利用
- SMTP AUTH
- TLS の利用

以下では、Java2SDK や Apache James が既にインストールされ、Apache James を利用できる環境が整っているものとして話を進める。テストで用いているのは、James-2.2.0 である。尚、設定の実際は主として Apache James のドキュメンテーション (以下、「ドキュメント」という) の参照によっている^(*)。

(*1)<http://james.apache.org/>上の Documentation を参照。

■テスト環境の構成

各種設定の動作を検証するにあたってのテスト環境についてまとめておく。

◆ホストマシン等

テスト環境下での、用いるホストマシンの構成は図1のとおりである。

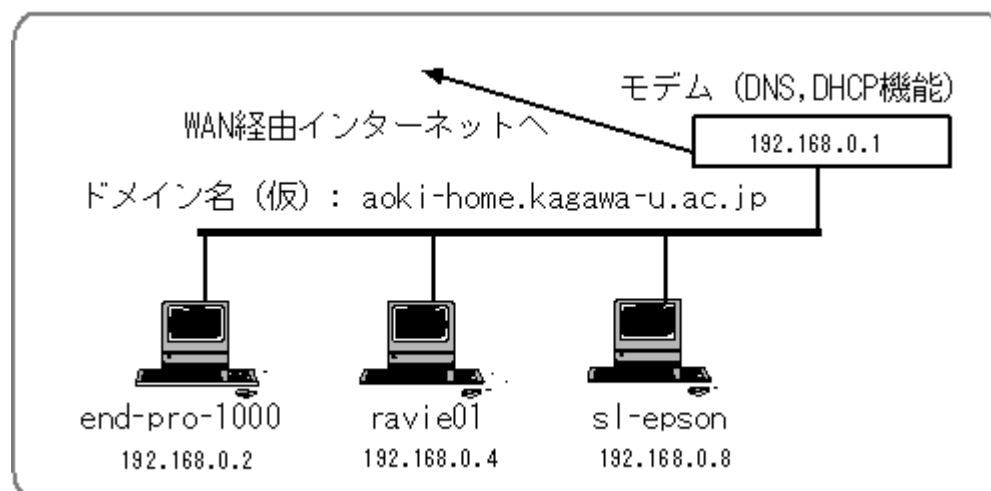


図1 : テスト環境のホスト構成

各ホストの IP アドレス等は次のようになっている。

end-pro-1000 ホスト

IP アドレス : 192.168.0.2

OS : Windows2000(SP4)と Linux MLD 7 (切り替えてブート可能)

DNS サーバの設定 : モデム(192.168.0.1)

ravie01 ホスト

IP アドレス : 192.168.0.4

OS : WindowsXP(SP2)

DNS サーバの設定 : モデム(192.168.0.1)

sl-epson ホスト

IP アドレス : 192.168.0.8

OS : FreeBSD-Release 4.9

DNS サーバの設定 : ローカルホスト(127.0.0.1)とモデム(192.168.0.1)

ISDN モデム

IP アドレス : 192.168.0.1

機能 : DNS, DHCP

ドメイン名 (仮) : aoki-home.kagawa-u.ac.jp

図1からも分かるように、3台のホストマシンはモデムに接続され、WANを経由して香川大学総合情報基盤センター幸町分室或はネットワークプロバイダーのPPPサーバに接続、インターネットへと繋がるようになっている。

テスト環境内にはDNSサーバは設定されていない。そのため、各ホストのhostsファイルにテスト環境下の全てのホストマシンについての情報を記述し、名前解決に当たってはhostsファイル情報が優先されるようにしている。モデムを通して外部に接続する際には、それぞれの接続先PPPサーバから接続先のDNS情報を得る。

◆Apache James のインストール先

Apache James は次のディレクトリ (フォルダ) にインストールされている。

Windows2000/XP ホスト : c:\james-2.2.0

FreeBSD, Linux MLD ホスト : /usr/local/james-2.2.0

以後、それぞれのインストール先ディレクトリを%JAMES_HOME%と表す。

◆登録済みテストユーザ

テストユーザとして各ホストにそれぞれ、ユーザ s-user01 と r-user01 を登録している。各ユーザのメールアドレスは、例えば ravie01 ホストの登録者であれば、s-user01@ravie01.aoki-home.kagawa-u.ac.jp, s-user01@[192.168.0.4] 等となる。

s-user01@[192.168.0.4]の表記は、メールの中継やメールリターンのテストに必要である。先にも述べたように、テスト環境下にはDNSサーバが設置されていない。一方 Apache James はメールを自ホスト外へ送信するに際しては、hosts ファイルを参照するのではなく、必ず送信先 SMTP サーバ名の MX レコードをDNSサーバに問い合わせる。そのため、テスト環境下でのメール中継等には、メールアドレスにSMTPサーバのIPアドレスが直接表記されていることが求められるのである。

■メーリングリストの設定

メーリングリストを設定する。簡単にメーリングリストを設定できるところが、Apache James の大きな特徴でもある。

◆設定の概要

メーリングリストの設定については、ドキュメントや文献(1)に詳しいが(*1)、これらはいずれも Apache James Ver.2.1 での設定である。ここでは、Ver.2.2.0 で加わったメーリングリストの設定法の紹介も兼ね、実際に設定する。この設定法についての詳しい説明がドキュメント等に与えられていないことから、やや詳しく紹介したい。

(*1)ドキュメント上の「How To Create Mailing Lists」や FAQ を参照。

●設定ファイルの存在等の確認

Apache James のインストールにより、メーリングリスト設定に関わる幾つかのファイルが与えられている。まず、それらのファイルや Apache James の設定ファイルである config.xml での関連する記述について見ておく。

james-listmanager.xml と james-liststores.xml

2つのファイルはいずれも %JAMES_HOME%/apps/james/conf/ に与えられている。james-listmanager.xml はメーリングリストに関わる Maillet パイプラインプロセスについてのサンプルであり、メーリングリスト・コマンドとメーリングリスト・メッセージを処理する 2つの Maillet プロセスの記述法が与えられている。ここで用いられている Matcher と Maillet プログラムは Ver.2.1 についてのドキュメントには無いもので、Ver.2.2 で提供されたものと考えられる。

james-liststores.xml はメーリングリストの登録者情報を保持するための設定ファイルのサンプルで、ここでの記述は Ver.2.1 でのものと同じである。

miResources.xml

これは、james-listmanager.xml 内で指定されているリソースファイルである。

config.xml における実体宣言

config.xml ファイルの先頭行近くに実体宣言に関わる記述があることに注意する（以下参照）。

```
<?xml version="1.0"?>
<!DOCTYPE config [
<!ENTITY listserverConfig SYSTEM "../conf/james-listmanager.xml">
<!ENTITY listserverStores SYSTEM "../conf/james-liststores.xml">
<!ENTITY fetchmailConfig SYSTEM "../conf/james-fetchmail.xml">
]>
```

また、この宣言に対応して、実体参照「&listserverConfig;」と「&listserverStores;」の記述が config.xml 内にコメントとして与えられていることを確認する。

●サンプルメーリングリストの設定

上記のサンプル設定ファイル等の確認から、メーリングリストの設定手順が次のとおりであることが分かる。

◆james-listmanager.xml と james-liststores.xml の 2つのファイルを編集する。

◆config.xml ファイル中の、「&listserverConfig;」と「&listserverStores;」をアンコメント

とする。

この手順に従い、サンプルとして与えられているメーリングリスト “testlist” を設定するとして、具体的な作業内容を見てみる。

james-listmanager.xml の編集とメーリングリストコマンド

少し長くなるが、james-listmanager.xml の内容をできるだけオリジナルな形で紹介する。もともとあるコメントを含め内容をよく読めばどの個所をどうすべきか分かるが、理解を助ける意味で、とりあえず修正すべき部分を分かりやすく太字で表している（以下参照）。

```

<!-- Add a Manager and Processor for each list -->
<!-- TODO: add more examples -->
<!-- CHECKME: check the <resource> path -->
  <mailet match="CommandListservMatcher="
    testlist@localhost" class="CommandListservManager"> ← (1)
    <!-- CHECKME! -->
    <!--resources-->
    <resources>/opt/james/apps/james/conf/miResources.xml</resources> ← (2)
    <!-- general information -->
    <listName>testlist</listName>
    <displayName>Test mailing list</displayName>
    <listOwner>owner@localhost</listOwner> ← (3)
    <repositoryName>list-test</repositoryName>
    <listDomain>localhost</listDomain> ← (4)
    <!-- the packages where the commands can be loaded from -->
    <commandpackages>
<commandpackage>org.apache.james.transport.mailets.listservcommands</commandpackage>
    </commandpackages>
    <!-- all the available commands for this command. -->
    <commands>
      <command name="subscribe" class="Subscribe"/>
      <command name="subscribe-confirm" class="SubscribeConfirm"/>
      <command name="unsubscribe" class="UnSubscribe"/>
      <command name="unsubscribe-confirm" class="UnSubscribeConfirm"/>
      <command name="error" class="ErrorCommand"/>
      <command name="owner" class="Owner"/>
      <command name="info" class="Info"/>
    </commands>
  </mailet>
  <!-- the processor for this list serv -->
  <mailet match="RecipientIs=testlist@localhost" class="CommandListservProcessor"> ← (5)
    <membersonly>>false</membersonly>
    <attachmentsallowed>>true</attachmentsallowed>
    <replytolist>>true</replytolist>
    <repositoryName>list-test</repositoryName>
    <subjectprefix>Test</subjectprefix>
    <autobracket>>true</autobracket>
    <listOwner>owner@localhost</listOwner> ← (6)
    <listName>testlist</listName>
  </mailet>

```

以下に(1)~(6)の部分について簡単に説明する。以下では end-pro-1000 ホストでの設定を仮定して記述しているので、環境に従って適宜読み替えてほしい。

(1) “localhost” の部分を “end-pro-1000.aoki-home.kagawa-u.ac.jp” とする。UNIX 系ホストの場合には “localhost” のみの記述でよいだろうが、Windows ホストの場合は丁寧に具体的に記述した方が無難である。以下の設定においても、“localhost” 部分は同様に書き

かえるものとする。尚，ここに記述されている

“CommandListservMatcher”と“CommandListservManager”

が，Ver.2.2.0で新しく導入されたMatcher-Mailletコンビネーションペアである。

(2)実際にApache Jamesがインストールされているディレクトリを記述する。ここでは，

/james2.2.0/apps/james/conf/miResources.xml

とする。

(3)メーリングリストtestlistのOwnerについて記述する。Ownerが仮にowner-tlistである
とすると，

owner-tlist@end-pro-1000.aoki-home.kagawa-u.ac.jp

となる。owner-tlistについては別途ユーザ登録を行っておく。

(4)，(5)“localhost”の部分を(1)と同様とする。(5)の部分の“CommandListservProcessor”
は新しく提供されたMailletである。

(6)Ownerについての記述なので，(3)と同様とする。

尚，上記ファイル内容の<commands>要素の記述から，メーリングリストコマンドとして次の
コマンドが用意されていることが分かる(*1)。

コマンド	内容
subscribe	メーリングリストに参加・登録する
subscribe-confirm	登録を確認する
unsubscribe	登録を取り消す
unsubscribe-confirm	登録の取り消しを確認する
error	エラーについて問い合わせる
owner	Owner宛にメールを出す
info	メーリングリストについての情報を得る

これらのコマンドの使用方法であるが，メールの宛先に
testlist-コマンド@end-pro-1000.aoki-home.kagawa-u.ac.jp
を指定してメールを送信する。

(*1)Ver.2.1では提供されているコマンドは **on** (登録をする) と **off** (登録の取り消し) の2つである。

james-liststores.xmlの編集

このファイルについては編集により特に変更を加える点はない。james-listmanager.xml
の内容との整合性をチェックするだけである (以下参照，太字の部分をチェックする)。

```

<!-- Mailing list repositories -->
<!-- Add a user repository for each list -->
<!-- TODO: add more examples -->
<repository name="list-test"
    class="org.apache.james.userrepository.UsersFileRepository">
    <destination URL="file://var/lists/list-test"/>
</repository>
    
```

この設定で，登録者のリストが%JAMES_HOME%/apps/james/var/lists/list-test/に保存さ
れる。

config.xmlの編集

ここでは，config.xml ファイル中で「&listserverConfig;」と「&listserverStores;」がコ

メントとされている部分を検索により見つけ出して、それらをアンコメントとする。
(以下太字の部分参照)。

```

<!-- remote delivery -->
  <processor name="transport">
    <!-- This is an example configuration including configuration for a list server.
-->
    <!-- CHECKME: before uncommenting this, edit the configuration file's
contents -->
    <!-- -->
      &listserverConfig;          ← コメントをはずす
    <!-- -->
===== 中 略 =====
    <!-- This is an example configuration including configuration for a list server. -->
    <!-- CHECKME: before uncommenting this, edit the configuration file's contents -->
    <!-- -->
      &listserverStores;        ← コメントをはずす
    <!-- -->
  </users-store>

```

◆設定の実際と動作確認

設定の粗筋が理解できたところで、実際にメーリングリスト設定を行い、動作を確認する。
設定は end-pro-1000 ホスト上である。

●設定の実際

メーリングリスト名を testlist とする。これまでに説明した内容に沿って関連ファイルを編集し、Apache James を再起動する。問題なく起動し、登録者のリストを保存する %JAMES_HOME%/apps/james/var/lists/list-test/ディレクトリ (フォルダ) が作成されていることを先ず確認する (図2参照)。

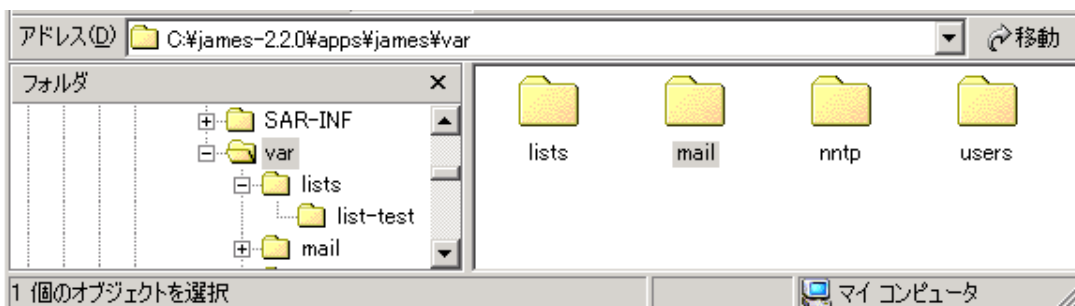


図2 : 作成された lists フォルダと list-test フォルダ

●メーリングリストへの登録

Owner の登録

先ず同じホスト内でメーリングリストの Owner である owner-tlist のユーザ登録を行っておく。この登録は、リストメンバーとしての登録ではなく、1 ユーザとしての登録である (登録方法については文献(1)を参照)。

メーリングリスト参加者の登録

同じホスト内のユーザ s-user01 をリストメンバーとして登録する。

送信者名 : s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp

宛先アドレス : testlist-subscribe@end-pro-1000.aoki-home.kagawa-u.ac.jp

のメールを作成，SMTP サーバに自ホストを指定してメールを送信する。このメール送信の手続きだけでは，リストへの参加者としての登録は未だなされていない。

上記メールに対して，owner-tlist 名で「登録確認のため，リプライメールを送信するよう」メールが返信されてくる（以下参照，便宜上編集している）。

```

===== 中 略 =====
From: Test mailing list <owner-tlist@end-pro-1000.aoki-home.kagawa-u.ac.jp>
Reply-To: testlist-subscribe-confirm@end-pro-1000.aoki-home.kagawa-u.ac.jp
To: s-user01 <s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp>
Subject: Please confirm your subscription to the testlist mailing list
===== 中 略 =====
Delivered-To: s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp

To confirm that you would like
  s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp
added to the testlist mailing list, please send an empty reply to this address:
  testlist-subscribe-confirm@end-pro-1000.aoki-home.kagawa-u.ac.jp
Usually, this happens when you just hit the "reply" button.
If this does not work, simply copy the address and paste it into the "To:" field of a new
message.
or click here:
  mailto:testlist-subscribe-confirm@end-pro-1000.aoki-home.kagawa-u.ac.jp
This confirmation serves two purposes. First, it verifies that I am able to get mail
through to you. Second, it protects you in case someone forges a subscription request in
your name.

```

このメールに対して返信して，リストメンバーとしての登録手続きが完了する。「登録手続き完了」を知らせるメール内容の概略を示す（以下参照，便宜上編集している）。

```

===== 中 略 =====
From: Test mailing list <owner-tlist@end-pro-1000.aoki-home.kagawa-u.ac.jp>
To: s-user01 <s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp>
Subject: Welcome to the testlist mailing list
===== 中 略 =====
Acknowledgment: I have added the address
  s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp
to the testlist mailing list.
Welcome to testlist@end-pro-1000.aoki-home.kagawa-u.ac.jp!
===== 以下 略 =====

```

この手続により，サーバでは登録済みメンバーをストアするためのファイルが作成される（図3）。

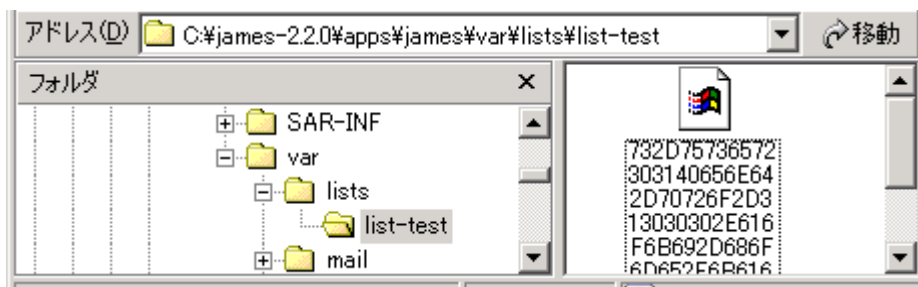


図3：作成されたリストメンバーをストアするためのファイル

同様にして，ravie01 ホストおよび sl-epson ホスト上のユーザ s-user01 をそれぞれ登録す

る。登録にあたっては、テスト環境下で返信メールが届くよう、それぞれ次のメールアドレスで登録した(*1)。

ravie01 ホスト上の s-user01 : s-user01@[192.168.0.4]

sl-epson ホスト上の s-user01 : s-user01@[192.168.0.8]

登録者を、リモートマネージャにより確認しておこう (以下参照)。

```

AMES Remote Administration Tool 2.2.0
Please enter your login and password
Login id:
Password:
Welcome root. HELP for a list of commands
user test-list                ← リポジトリを test-list に変更
Changed to repository 'list-test'.
listusers                    ← 登録者を確認
Existing accounts 3
user: s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp
user: s-user01@[192.168.0.4]
user: s-user01@[192.168.0.8]

```

(*1)このように設定する理由については上述したとおりであるが、詳細については文献(1)を参照のこと。通常の設定下では、このように IP アドレスを用いたメールアドレスを用いる必要はない。

●メールングリストコマンドへの応答確認

各種コマンドへの応答を確認する。メールの宛先アドレスは、コマンドを `command` として

`testlist-command@end-pro-1000.aoki-home.kagawa-u.ac.jp`

である。各コマンドへの応答内容は以下に示すとおりである。

info

メールングリストの **Owner** である `owner-tlist` からのメールが届く。内容は、メールングリストへの投稿の方法、届いたメールへの返信 (リプライ) の方法、登録者のリスト、各種コマンドの使い方等である。尚、未登録者から送信された場合にも、同内容のメールが届けられる。登録者専用ではない。

error

エラーについての報告メールが **Owner** より届く。例えば、`User Unknown` のメールを送信したなどの内容である。

owner

Owner 宛のメールで、**Owner** 個人に配信される。

unsubscribe

メールを送信すると、「確認のため、`unsubscribe-confirm` コマンドによるメールを返信するよう」とのメールが **Owner** より届く。尚、未登録者からの送信に対しては、「`Invalid request`」の題名でメールが返信されてくる。

unsubscribe-confirm

メールを送信すると、「`Goodbye from the testlist mailing list`」の題名で、登録者のリストからアドレスを取り除いた旨のメールが **Owner** より届く。

●メールングリストへの投稿

sl-spson ホストのユーザ s-user01@[192.168.0.8]からメーリングリストに投稿し、各登録ユーザ宛にメールが配信されることを確認する。メールは登録者全てに配信された。
 ravie01 ホストのユーザ s-user01@[192.168.0.4]宛に配信されたメールの内容を示す（下記参照）。

```
Return-Path: <owner-tlist@end-pro-1000.aoki-home.kagawa-u.ac.jp> ← (1)
===== 中 略 =====
Date: Mon, 22 Nov 2004 08:15:37 +0900
From: S-USER01 <s-user01@[192.168.0.8]> ← (2)
===== 中 略 =====
To: testlist@end-pro-1000.aoki-home.kagawa-u.ac.jp
Subject: [Test]メーリングリストへの投稿01 ← (3)
===== 中 略 =====
Reply-To: testlist@end-pro-1000.aoki-home.kagawa-u.ac.jp ← (4)
Delivered-To: s-user01@[192.168.0.4]

メーリングリストへの投稿01
01-Mail
2004.11.22

-----
To unsubscribe, e-mail: testlist-unsubscribe@end-pro-1000.aoki-home.kagawa-u.ac.jp
For additional commands, e-mail: testlist-info@end-pro-1000.aoki-home.kagawa-u.ac.jp
```

配信されたメールのヘッダ内容に注目すると以下のようにになっている。

- (1)ヘッダの Return-Path はメーリングリストの Owner 宛である。
- (2)ヘッダの From: はメールの投稿者アドレスである。
- (3)ヘッダの Subject: には、元の Subject 内容に Prefix として[Test]が付け加えられている。これは、james-listmanager.xml において、
 <subjectprefix>Test</subjectprefix>, <autobracket>>true</autobracket>
 となっていることによる（下記(b), (c)を参照,.)。

```
<mailet match="RecipientIs=testlist@localhost" class="CommandListservProcessor">
===== 中 略 =====
    <replytolist>>true</replytolist> ← (a)
    <repositoryName>list-test</repositoryName>
    <subjectprefix>Test</subjectprefix> ← (b)
    <autobracket>>true</autobracket> ← ©
    <listOwner>owner@localhost</listOwner>
    <listName>testlist</listName>
</mailet>
```

(4)ヘッダの Reply-To: にはメーリングリストのアドレスが与えられている（上記の(a)を参照）。従って、配信されたメールにリプライすると、それはメーリングリスト宛となる。

◆その他の設定

●Ver.2.1 タイプのメーリングリストとの併用

Ver.2.1 で提供されているメーリングリストをわざわざ利用することもないのであるが、2つの方法を併用する場合の設定について説明する。作成するメーリングリスト名を kyulist とする。

james-listmanager-v21.xml の作成

james-listmanager.xml とは別に james-listmanager-v21.xml ファイルを作成する。内容

は次の様である。

```

<!-- Add a Manager and Processor for each list -->
<mailet
match="CommandForListserv=kyulist@end-pro-1000.aoki-home.kagawa-u.ac.jp"
class="AvalonListservManager">
  <repositoryName>list-kyu</repositoryName>
</mailet>
<!-- the processor for this list serv -->
<mailet match="RecipientIs=kyulist@end-pro-1000.aoki-home.kagawa-u.ac.jp"
class="AvalonListserv">
  <membersonly>>false</membersonly>
  <attachmentsallowed>>true</attachmentsallowed>
  <replytolist>>true</replytolist>
  <repositoryName>list-kyu</repositoryName>
  <subjectprefix>KYU</subjectprefix>
  <autobracket>>true</autobracket>
</mailet>

```

james-liststores.xml への追加記述

james-liststores.xml に kyulist への登録者リストをストアするための記述を追加する。追加する内容を示す（以下参照）。

```

<!--
  <repository name="list-kyu"
    class="org.apache.james.userrepository.ListUsersJdbcRepository"
    destinationURL="db://maildb/lists/list-kyu">
    <sqlFile>file://conf/sqlResources.xml</sqlFile>
  </repository> -->
  <repository name="list-kyu"
    class="org.apache.james.userrepository.UsersFileRepository">
    <destination URL="file://var/lists/list-kyu"/>
  </repository>

```

config.xml への追加記述

james-listmanager-v21.xml を参照するよう記述を追加する。実体宣言の個所を

```

<!DOCTYPE config [
<!ENTITY listserverConfig SYSTEM "../conf/james-listmanager.xml">
<!ENTITY listserverConfigV21 SYSTEM "../conf/james-listmanager-v21.xml">
===== 中 略 =====
]>

```

とし、root プロセッサの記述部分の末尾近くに「&listserverConfigV21;」を挿入する（以下参照）。

```

<processor name="root">
===== 中 略 =====
  <b>&listserverConfigV21;</b>
<!-- Send remaining mails to the transport processor for either local or remote delivery -->
  <mailet match="All" class="ToProcessor">
    <processor> transport </processor>
  </mailet>
</processor>

```

以上で、メーリングリストへの参加登録、登録の取り消し、投稿は
 kyulist-on@end-pro-1000.aoki-home.kagawa-u.ac.jp ← 参加登録
 kyulist-off@end-pro-1000.aoki-home.kagawa-u.ac.jp ← 登録取り消し

kyulist@end-pro-1000.aoki-home.kagawa-u.ac.jp ← 投稿

宛にメールを送信することにより行える(*1)。

(*1)上述したように、Ver.2.1 では **on** (登録をする) と **off** (登録の取り消し) の2つのコマンドだけが提供されている。したがって、メーリングリストへの登録・登録取り消しにおいては、確認を求めるメールは配信されない。登録の際には「Successfully added to listserv.」、登録取り消しでは「Successfully removed from listserv.」を内容とするメッセージが配信されるのみである。

●リソースファイル miResources.xml のカスタマイズ

Ver.2.2 で提供されているメーリングリストでは、メーリングリストコマンドの処理に関わる設定は上述したように james-listmanager.xml ファイルの編集による。そこでリソースファイルとして指定されているのが miResources.xml ファイルであり、このファイルの編集により、コマンドへの応答メールの内容をカスタマイズすることができる。応答として送信するメールの内容をより詳しく、或いはより簡潔にする場合に設定を変更して用いるとよい(*1)。

他のリソースファイルを用いる

miResources.xml ファイルのコピーを別名 (例えば、miResources2.xml) で保存し、そのファイルを適宜編集した上で、james-listmanager.xml でのリソースファイル指定を変更する (以下参照)。

```
<mailet
match="CommandListservMatcher=testlist@end-pro-1000.aoki-home.kagawa-u.ac.jp"
class="CommandListservManager">          ← (1)
  <!-- CHECKME! -->
  <!--resources-->
  <resources>/james-2.2.0/apps/james/conf/miResources2.xml</resources>
===== 以下 略 =====
```

カスタマイズ例

メーリングリストへの登録者数が多くなってきた場合、info コマンドへの応答メールの中で「現在の登録者」全てをリストアップすることは適切ではないであろう。登録者をリストアップしないように変更する場合の例を示す (以下参照、太字の部分を削除)。

```
===== 中 略 =====
<group name="info">
  <resource name="info.subject">Information regarding the ${LIST_NAME} mailing
list</resource>
  <resource name="text" xml:space="preserve"><![CDATA[
===== 中 略 =====
--- Current members ---
The list of current members are:
  }${MEMBER_LIST}]]>          ← CDATA セクションの内容を編集する
  </resource>
</group>
```

footer グループや header グループなど、それぞれのグループ内の主として CDATA セクション部分を中心に編集すれば、応答メールの内容を好みに応じたものとすることができる。

(*1)応答メール内容を日本語で表すことについてコメントしておく。現段階では、応答メール作成において日本語には対応していないように思われる。

まず、リソースファイルの CDATA セクションなどに日本語の記述を挿入する場合、文字エンコーディングが UTF-8 以外の場合には、miResources.xml ファイルの先頭行に、例えば

```
<?xml version="1.0" encoding="Windows-31J"?>
```

のように、用いるエンコーディング名を記述した行を挿入しておかなければ、起動時にエラーとなる。

次に、上のように文字エンコーディングを指定して起動がうまく行った場合でも、返答されるメールにおいて「????」のように日本語の文字列部分に文字化けが発生する。

■データベース MySQL の利用

Apache James では JDBC (Java DataBase Connectivity) によるデータベースとの連携が可能である。特に、MySQL データベースを利用するについては、JDBC ドライバが含まれてインストールされており、また、`config.xml` ファイル内に関連する設定がコメントされた形で与えられている。コメントを外せば少しの修正を加えることで MySQL の利用が可能である。

◆MySQL の入手とインストール

テスト環境下の各ホストマシンには、他の Web アプリケーション作成の関係で既に MySQL を入手あるいはインストールしており、以下のテストではそれらを利用するが、簡単にフィルの入手とインストール方法についてまとめておく。

●MySQL の入手

以下のサイトからダウンロードが可能である。

<http://www.mysql.com/>

各 OS に適したファイルをダウンロードする。本稿執筆時点 (2004 年 11 月 26 日現在) での最新バージョンは 4.1 である。

●インストール

Windows2000/XP

既にインストールされているのは MySQL Ver.4.0.17 で、インストールに際して入手したファイルは `mysql-4.0.17-win.zip` である。適当なフォルダ内で zip ファイルを展開し、作成されたフォルダ内の SETUP プログラムを実行すればインストールされるとともに、サービスにも登録される。

Linux MLD

`mysql-standard-4.0.17-pc-linux-i686.tar.gz` ファイルを用いるとすれば、`/usr/local/`などで展開すればよい (以下参照)。

```
# tar zxvf mysql-standard-4.0.17-pc-linux-i686.tar.gz
```

或いは、rpm ファイルを用いてインストールすればより簡便である。

FreeBSD

`mysql-standard-4.0.17-unknown-freebsd4.7-i386.tar.gz` などのファイルを用いて展開してもよいが、本テスト環境下のホストでは Ports コレクションからインストールを行っている (Ver.4.1.0-alpha, 以下参照)。尚、パッケージからのインストールだと一層簡便である。

```
%pwd
/usr/ports/databases/mysql41-server
%make
%su
Password:
sl-epson# make install
```

◆MySQL の利用設定

Apache James で MySQL ができるよう設定を行う。

●3つのデータ保存タイプ

MySQL の利用設定に先立って、Apache James で実装されている各種データを保存するタイプについて説明する。3つのタイプが利用可能である。

File リポジトリ

データをファイル形式で保存するタイプで、デフォルトでは全てのデータ（メールメッセージデータ、ユーザ情報データ等）がこのタイプで保存されるよう設定されている。これまでに説明した（「james-liststores.xml への追加記述」を参照）

```
<repository name="list-kyu"
  class="org.apache.james.userrepository.UsersFileRepository">
  <destination URL="file://var/lists/list-kyu/">
</repository>
```

において、

```
destinationURL="file://var/lists/list-kyu/"
```

のようにプレフィックス「file:」で始まる記述が、この File リポジトリタイプであることを示している。

Database(JDBC) リポジトリ

データをデータベースに保存するタイプで、メールメッセージやユーザ情報の保存、メール処理過程中での Spool にデータベースを利用することができるよう実装されている。ただし、デフォルトでは config.xml ファイルの中で全てコメント扱いとなっている。

例を示すと、

```
<!-- <repository name="list-kyu"
  class="org.apache.james.userrepository.ListUsersJdbcRepository"
  destinationURL="db://maildb/lists/list-kyu">
  <sqlFile>file://conf/sqlResources.xml</sqlFile>
</repository> -->
```

において、

```
destinationURL="db://maildb/lists/list-kyu"
```

のように、プレフィックス「db:」で始まる記述が Database(JDBC) リポジトリタイプである。

DBFile リポジトリ

メールの保存にのみ用いられる特殊なタイプで、メールメッセージの本体部分をファイル形式で保存し、メールのヘッダ情報をデータベースに保存する。「dbfile:」のプレフィックスを用いて記述される。

●MySQL との接続確立設定

Apache James から JDBC を通じて MySQL との接続が可能ないように設定する。Apache James には MySQL 用の JDBC ドライバがインストール済みである。

MySQL 側での mail データベースの作成

デフォルトで与えられている config.xml ファイルでは、MySQL 側にデータベースとして mail が作成されていることが前提とされており、それを利用する形となっている（勿論他のデータベース名を用いることも可能である）。ここでは、config.xml に予め与えられているように mail データベースを作成する（以下参照、適宜編集している）。

```
C:¥mysql¥bin>mysql
Welcome to the MySQL monitor.  Commands end wit
```



```

Your MySQL connection id is 5 to server version
Type 'help;' or ' h' for help. Type ' c' to cle
mysql> create database mail;           ← データベース mail を作成
Query OK, 1 row affected (0.01 sec)
mysql> show databases;
+-----+
| Database |
+-----+
| mail     |
| my_db    |
| mysql    |
| test     |
+-----+
4 rows in set (0.00 sec)
mysql> use mail                       ← データベース mail を選ぶ
Database changed
mysql> show tables;                   ← テーブルを表示
Empty set (0.00 sec)
mysql> quit
Bye

```

config.xml ファイルでの設定

config.xml ファイルでは、“database-connections” ブロックでコメント扱いとなっている MySQL に関する部分をアンコメントとする（以下参照）。

```

<!-- -->
  <data-source name="maildb"
    class="org.apache.james.util.dbcp.JdbcDataSource">
    <driver>org.git.mm.mysql.Driver</driver>
    <dburl>jdbc:mysql://127.0.0.1/mail?autoReconnect=true</dburl>
    <user>user</user>           ← ユーザ名は架空の user とした
    <password></password>     ← パスワードは空とした
    <max>20</max>
  </data-source>
<!-- -->

```

上の設定で、接続名“maildb”でMySQL内のデータベースmailとの接続が可能となる。

Apache James の起動確認

以上の設定で、エラーを生じることなく Apache James が起動すれば、MySQL との連携の第1ステップはクリアできたこととなる。問題なく起動することが確認できるはずである。

◆MySQL の利用例

MySQL との連携がうまく行くことを確認するために、ユーザ情報を MySQL に格納することを試みる。

●ローカルユーザ情報の MySQL への格納

リポジトリタイプの変更

config.xml ファイルの“The User Storage block”部分で、リポジトリのタイプを File タイプから Database(JDBC) タイプに切り替える。具体的には、File タイプとする記述部分をコメント扱いとし、Database(JDBC) タイプとする部分をアンコメントとする（以下参照）。

```

<!--
  repository name="LocalUsers"
  class="org.apache.james.userrepository.UsersFileRepository">
  <destination URL="file://var/users/">

```

```

</repository -->          ← この部分をコメント扱いにする
<!-- Database backed user repositories -->
===== 中 略 =====
<!-- -->                  ← コメントとする
  <repository                                name="LocalUsers"
class="org.apache.james.userrepository.JamesUsersJdbcRepository"
destinationURL="db://maildb/users">
  <sqlFile>file://conf/sqlResources.xml</sqlFile>
</repository>
<!-- -->                  ← コメントとする

```

上の設定で、MySQL内でデータベース名が mail、テーブル名が users にユーザの登録情報が格納される。

ローカルユーザの登録

ローカルユーザとして、s-user01, r-user01, postmaster を登録し、postmaster のエイリアスを s-user01 に設定する。登録はリモートマネージャを用いて行うが、登録方法や登録状況の確認等については文献(1)を参照して欲しい。ここでは、リモートマネージャではなくMySQLにより登録状況を直接確認する（以下参照、適宜編集している）。設定したとおりにMySQLにユーザが登録されていることが分かる。

```

mysql> connect mail          ← データベース mail を選ぶ
Connection id: 38
Current database: mail
mysql> show tables;        ← テーブルの表示
+-----+
| Tables_in_mail |
+-----+
| users          | ← user テーブルが作成されている
+-----+
1 row in set (0.00 sec)
mysql> select username,alias from users; ← テーブルの項目を表示
+-----+-----+
| username  | alias  |
+-----+-----+
| s-user01  |       |
| r-user01  |       |
| postmaster| s-user01 |
+-----+-----+
3 rows in set (0.00 sec)
mysql>

```

メールの送信と受信

メールの送信とPOP3による受信を確認するため、s-user-01 から r-user01 宛にテストメールを送信し、POP3を用いて受信する（以下参照）。MySQLに格納されたユーザ情報が適切に利用されていることが確認できる。

```

Return-Path: <s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp>
===== 中 略 =====
To: r-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp
Subject: DB テストメール, 2004.11.27
===== 中 略 =====
Delivered-To: r-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp

DB テストメール, 2004.11.27

```

●メールリングリスト登録者の MySQL への格納

リポジトリタイプの変更

例として、前述の「■メールリングリストの設定」の「◆その他の設定、●Ver.2.1 タイプのメールリングリストとの併用」で紹介した Ver.2.1 タイプのメールリングリスト “kyulist” の設定を利用する。「james-liststores.xml への追加記述」で示したリポジトリ設定でコメント扱いとなっている Database(JDBC)タイプの設定をアンコメントとし、File タイプの設定部分をコメントとする。リポジトリ名は “list-kyu” である。Apache James 再起動後のリポジトリ作成状況を先ず確認する。リモートマネージャでの確認では（以下参照）

```
JAMES Remote Administration Tool 2.2.0
Please enter your login and password
Login id:
Password:
Welcome root. HELP for a list of commands
user list-kyu           ← リポジトリを list-kyu に変更
Changed to repository 'list-kyu'.
listusers           ← 登録者を表示
Existing accounts 0     ← 現段階では登録者なし
```

となり、また MySQL での直接確認では次の様となる（以下参照）。

```
mysql> connect mail           ← データベース mail を選ぶ
Connection id: 15
Current database: mail

mysql> show tables;         ← テーブルを表示
+-----+
| Tables_in_mail |
+-----+
| lists          | ← lists テーブルが作成されている
| users          |
+-----+
2 rows in set (0.02 sec)
mysql> select * from lists; ← lists テーブルの項目を表示
Empty set (0.00 sec)       ← 現段階では登録者なし
mysql>
```

メールリングリストへの登録

前に説明したように、メールリングリストへの登録は以下のアドレスにメールを送信することによる。

メールの送信先アドレス : kyulist-on@end-pro-1000.aoki-home.kagawa-u.ac.jp

次のユーザを実際に登録してみる。

s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp

s-user01@[192.168.0.4] , s-user01@[192.168.0.8]

各登録者宛に “Successfully added to listserv.” を内容とするメールが届けば登録完了である。MySQL に登録情報が格納されていることを確認する（以下参照）。

```
==== MySQL での直接確認 ====
mysql> select * from lists;
+-----+-----+-----+
| listName | listSubscriber |
+-----+-----+-----+
| list-kyu | s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp |
```

```

| list-kyu | s-user01@[192.168.0.4] |
| list-kyu | s-user01@[192.168.0.8] |
+-----+
3 rows in set (0.00 sec)
===== リモートマネージャによる確認 =====
Welcome root. HELP for a list of commands
user list-kyu
Changed to repository 'list-kyu'.
listusers
Existing accounts 3
user: s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp
user: s-user01@[192.168.0.4]
user: s-user01@[192.168.0.8]

```

以上で MySQL との連携がうまく設定できていることが確認できた。メーリングリストのように登録者数が大きくなる場合など、リポジトリとして File タイプではなく、Database(JDBC)タイプを選べば効率的であろう。spool や inboxes など他のリポジトリについても、config.xml ファイルに MySQL を利用する設定がコメントとして与えられているので、適宜状況に応じて MySQL を利用すればよい。

■SMTP AUTH

SMTP AUTH(Authenticated SMTP, SMTP Authentication, RFC2554)は、SMTP クライアントからの接続に際して、クライアント側のユーザアカウントとパスワードを用いて認証し、もって通常は許可されていない他の SMTP サーバへのメール中継を許可しようというものである^③。スパマーによるメール中継の踏み台 (Open Relays) とならないための方法として、POP before SMTP やロームユーザ登録による方法^④ ^⑤に替わり現在普及しつつあるところである。

Apache James は SMTP AUTH 機能を実装しており、config.xml ファイルでの設定により、容易にこの機能を実現できる。

◆SMTP AUTH の設定

●設定の概要

ドキュメントに沿いつつ、config.xml ファイルにおける設定の概要を紹介する。要点は以下の4点である。

(1)SMTP コネクションの時点で、メールが自ホスト宛であるか外部への中継を要するものであるかを判別する必要がある。自ホスト宛メールとして受け取るべきドメインを config.xml ファイルにおいて明示しておかねばならない。これは、James ブロックの<servername>要素において設定する。既に設定済みである。文献(1)を参照のこと。

(2)SMTP AUTH は、認証により任意の外部 SMTP クライアントからのメール中継を許可しようというものである。このことから、メール中継の踏み台とならないよう中継を許可するクライアント側のネットワークアドレスを指定している設定、transport プロセッサブロック内の “RemoteAddrNotInNetwork” Maillet の設定、をコメントとするか削除しておく。この設定についても文献(1)を参照のこと。

(3)config.xml ファイルの SMTP サーバについての設定ブロックにおいて、SMTP AUTH の設定を “true” とする。

(4)認証された接続者の送信者メールアドレスが、接続者自身のものである場合に限り中継を許可するのであれば、<verifyIdentity>要素を“true”に設定する。メールアドレスの詐称を防ぐ意味からも設定しておく方がよい。

以上の設定が終了したならば、Apache James を再起動すればよい。

●具体的な設定

以下の設定とその検証テストでは、検証の都合上、SMTP クライアントとして end-pro-1000 ホスト、SMTP サーバとして ravie01 ホストを用い、中継先サーバとして sl-epson ホストを用いる (図4参照)

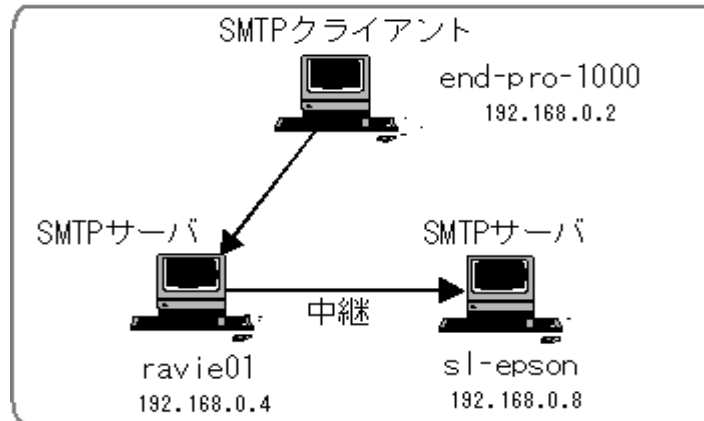


図4：検証テストの構成

ravie01 ホストの config.xml ファイルでの具体的な設定内容 (太字の部分) を次に示す (以下参照)。

```

<!--The SMTP server is enabled by default -->
<smtpserver enabled="true">
  <port>25</port>
  <!-- bind> </bind -->           ← コメントのまま
  <!-- useTLS>true</useTLS -->   ← コメントのまま
  <handler>
    <helloName autodetect="true">ravie01.aoki-home.kagawa-u.ac.jp</helloName>
    <connectiontimeout>360000</connectiontimeout>
    <authRequired>true</authRequired> ← コメントを外し“true”とする
    <!-- authorizedAddresses>127.0.0.0/8</authorizedAddresses -->
    <authorizedAddresses>127.0.0.0/8, 192.168.0.4/32</authorizedAddresses>
    <verifyIdentity>true</verifyIdentity> ← コメントを外し“true”とする
    <maxmessagesize>0</maxmessagesize>
  </handler>
</smtpserver>
  
```

上の設定での、

```
<authorizedAddresses>127.0.0.0/8, 192.168.0.4/32</authorizedAddresses>
```

については後ほど採り上げる。このタグ要素の<authRequired>>false</authRequired> (デフォルト設定、SMTP AUTH 未使用時) の場合における役割については文献(1)を参照していただくとして、ここでの設定は、他ホスト (自ホスト以外) からの接続によるメール中継は、認証手続を経ない限り、SMTP コネクションレベルにおいて許可しない設定であることを指摘しておく。

◆SMTP AUTH 設定の検証

●SMTP AUTH 設定と認証方式の確認

Apache James の認証方式を調べるため、end-pro-1000 ホストから ravi01 の 25 番ポートに telnet で接続する。次の結果から Apache James がサポートする認証方式は LOGIN と PLAIN であることが分かる（以下参照）(*1)。

```
C:\Documents and Settings\Administrator>telnet ravi01 25
220 ravi01 SMTP Server (JAMES SMTP Server 2.2.0) ready Mon, 29 Nov 2004 05:25:34 +0900 (JST)
ehlo end-pro-1000
250-ravi01 Hello end-pro-1000 (end-pro-1000.aoki-home.kagawa-u.ac.jp [192.168.0.2])
250-AUTH LOGIN PLAIN
250 AUTH=LOGIN PLAIN
quit
221 ravi01 Service closing transmission channel
ホストとの接続が切断されました。
C:\Documents and Settings\Administrator>
```

上記の 2 行

250-AUTH LOGIN PLAIN

250 AUTH=LOGIN PLAIN

が表示されるのは、end-pro-1000 ホストが

```
<authorizedAddresses>127.0.0.0/8, 192.168.0.4/32</authorizedAddresses>
```

の記述に含まれていないからで、この 2 行の表示に反応して、クライアント側が AUTH コマンドをサーバに送信し（クライアント側の設定により、送信しないことも可能）、認証手続きがとられることとなる（文献(3)を参照）。因みに ravi01 ホスト自身から 25 番ポートに接続した場合には、上の 2 行は表示されない。

●予備的な確認テスト

end-pro-1000 ホスト上のメーラを用い、SMTP サーバに ravi01 を指定して、ravi01 のユーザ宛等にメールを送信する。メーラ側での設定は、通常の SMTP AUTH を利用しないままであるとする(後出の図6で、「名前とパスワードを使用する」をチェックしない設定)。したがって、メーラは

250-AUTH LOGIN PLAIN

が返されても、パスワード入力のためのプロンプトを表示せず、したがって AUTH コマンドを送信しない。

 ravi01 ホストユーザ宛のメール送信

送信者アドレス：s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp

(end-pro-1000 ホストに登録されているユーザ)

受信者アドレス：r-user01@ravi01.aoki-home.kagawa-u.ac.jp

(ravi01 ホストのユーザ)

としてメールを送信する。

メーラ側から AUTH コマンドを送信せず認証手続きがスキップされているが、メールは自ホストユーザ宛であることからサーバにより受信された。

中継メールの送信

送信者アドレス : s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp
(end-pro-1000 ホストのユーザ)

受信者アドレス : r-user01@[192.168.0.8] (sl-epson ホストのユーザ)

としてメールを送信し、認証手続きがスキップされた場合にはメール中継が許可されないことを確認する。これは

<authorizedAddresses>127.0.0.0/8, 192.168.0.4/32</authorizedAddresses>

の設定の確認である。メール中継は、認証手続きが必要であるとして拒否された (図5)。



図5 : メール中継が拒否されたことを示す警告

以上のテストから、サーバ側で SMTP AUTH が設定されている場合には、自ホストあてのメールは認証手続きが採られなくても受信され、他サーバへのメール中継は認証が無ければ拒否されることがあらためて確認された。

●メールの送信テスト

認証方式が LOGIN と PLAIN であることが分かったので、メーラとして PLAIN 認証をサポートする Netscape に付属のメーラを用いてメール送信のテストを行う(*1)。

メーラの設定

end-pro-1000 ホスト上のメーラにおける SMTP サーバの設定では、図6に示すように「名前とパスワードを使用する」項目にチェックマークを入れ、名前を設定する。

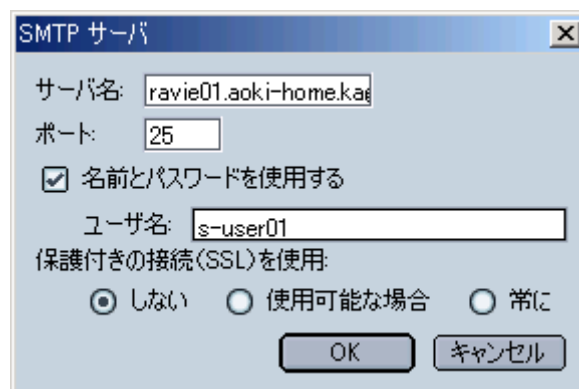


図6 : メーラでの SMTP サーバの設定

メールの送信

送信者アドレス : s-user01@ravie01.aoki-home.kagawa-u.ac.jp

(ravie01 ホストに登録されているユーザアドレスを用いる)

受信者アドレス : r-user01@[192.168.0.8] (sl-epson ホストのユーザ)

としてメーラを用いてメールを送信する(*2)。

メーラの SMTP サーバへの接続に際してパスワードの入力が求められ (図7参照、この後

メーラからサーバに AUTH コマンドが送信される), パスワードの認証が通ったのちメールが受け付けられた。

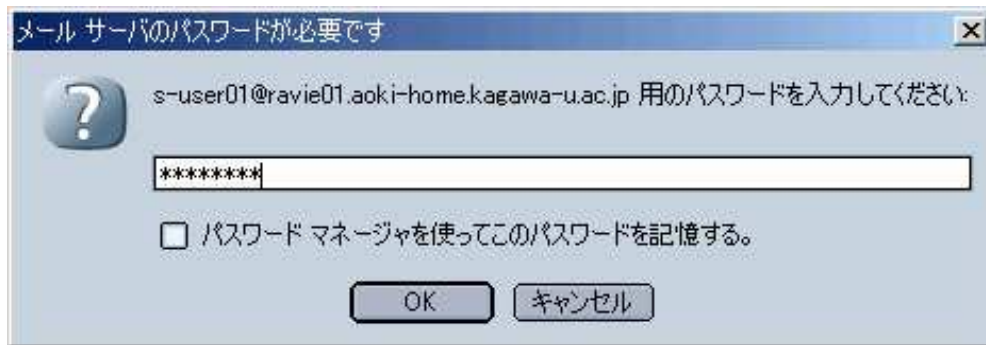


図7：パスワードの入力画面

中継されたメールの確認

sl-epson ホストで受信したメールの内容を簡単に示す。中継されたことが分かる程度の内容であり、Sendmail における SMTP AUTH のように、認証されたことを示すヘッダ部分の記述はない（以下参照）。

```
Return-Path: <s-user01@ravie01.aoki-home.kagawa-u.ac.jp>
Received: from ravie01.aoki-home.kagawa-u.ac.jp ([192.168.0.4])
        by sl-epson.aoki-home.kagawa-u.ac.jp (JAMES SMTP Server 2.2.0) with
SMTP ID 611
        for <r-user01@[192.168.0.8]>;
        Mon, 29 Nov 2004 08:47:03 +0900 (JST)
Received: from end-pro-1000.aoki-home.kagawa-u.ac.jp ([192.168.0.2])
        by ravie01 (JAMES SMTP Server 2.2.0) with SMTP ID 28
        for <r-user01@[192.168.0.8]>;
        Mon, 29 Nov 2004 08:30:30 +0900 (JST)
Message-ID: <41AA5E57.2060303@ravie01.aoki-home.kagawa-u.ac.jp>
Date: Mon, 29 Nov 2004 08:25:11 +0900
From: S-USER01 <s-user01@ravie01.aoki-home.kagawa-u.ac.jp>
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; ja-JP; rv:1.4) Gecko/20030624
Netscape/7.1 (ax)
X-Accept-Language: ja
MIME-Version: 1.0
To: r-user01@[192.168.0.8]
Subject: SMTP AUTH 10
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
Delivered-To: r-user01@[192.168.0.8]

SMTP AUTH 10
```

ravie01 ホストに残されたログから

ravie01 ホストに残されたログから、r-user01@[192.168.0.8]宛のメールを受信し中継した際の関係部分を示す（太字の部分に注目、以下参照）。

```
===== smtpserver についてのログ =====
29/11/04 08:30:22 INFO  smtpserver: SMTP Service started plain:25
29/11/04 08:30:30 INFO  smtpserver: Connection from
end-pro-1000.aoki-home.kagawa-u.ac.jp (192.168.0.2)
29/11/04 08:30:30 INFO  smtpserver: AUTH method PLAIN succeeded
29/11/04 08:30:30 INFO  smtpserver: Successfully spooled mail Mail1101684630686-0
from s-user01@ravie01.aoki-home.kagawa-u.ac.jp for [r-user01@[192.168.0.8]]
```



```

===== maillet 関連のログ =====
29/11/04 08:31:31 INFO James.Maillet: RemoteDelivery: Attempting delivery of
Mail1101684630686-0-to-192.168.0.8 to host 192.168.0.8 at 192.168.0.8 to addresses
[<r-user01@[192.168.0.8]>]
29/11/04 08:31:32 INFO James.Maillet: RemoteDelivery: Mail
(Mail1101684630686-0-to-192.168.0.8) sent successfully to 192.168.0.8 at 192.168.0.8

```

上のログ記録からも、認証方式が PLAIN であったことがはっきりする。ただ、PLAIN 認証では、ほぼ生の形でユーザの名前やパスワードが流れており、セキュリティの面で十分ではない。

(*1)各メーラがどのような認証方式に対応しているのかについては、現時点では少し古いデータとなっ
てしまっているが、文献(3)を参照してほしい。

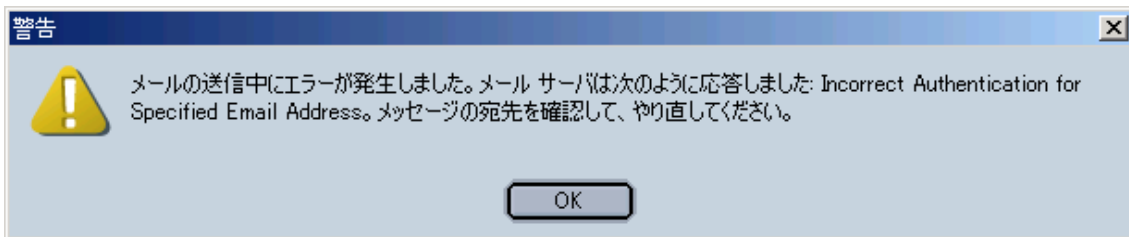
(*2)送信者アドレスのドメイン部分を詐称した場合を例にとり、

```
<verifyIdentity>true</verifyIdentity>
```

の設定の意味を確認しておく。送信者アドレスを

```
s-user01@fake.domain.jp
```

としてメールを送信する。宛先は ravie01 ホストユーザでも中継を要する sl-epson ホストユーザであ
ってもよい。現設定では、メーラは SMTP AUTH コマンドを送信する。次の図のような警告が表示
されメールの受け付けそのものが拒否された。メールアドレスは、メールサーバに登録された正しい
ものでなければメールは受け付けられない。



■ TLS の利用

通信路を暗号化するため、TLS (Transport Layer Security, RFC 2246) を利用する設定を行う。ドキュメントの「James2.1-Using TLS (usingTLS_2_1.html)」の内容に沿って設定作業を進める。

◆ TLS 利用の設定

Apache James では、TLS の利用にあたって JSSE (Sun Java Secure Sockets Extension) が必要である。もし、利用環境において JSSE が導入されていない場合には、先ず JSSE をダウンロードし、利用できるよう設定する必要がある。このテストで用いている Java2SDK-1.4.x にはパッケージとして JSEE が同梱されており、以下では JSSE の利用が可能となっていることを前提として話を進める。

● config.xml ファイルでの設定内容

Apache James での TLS 利用にあたって必要な設定が何かを知るには、config.xml ファイル内での該当する記述を眺めてみるのが手っ取り早い。関連するのは、“Socket Manager block”のコメント扱いとなっている<server-sockets>要素である (以下参照)。

```

<sockets>
  <server-sockets>
===== 中 略 =====
  <!--

```

```

<factory name="ssl"
class="org.apache.avalon.cornerstone.blocks.sockets.TLSServerSocketFactory">
  <keystore>
    <file>conf/keystore</file>          ← Keystore ファイルの path
    <password>secret</password>        ← Keystore ファイルのパスワード
    <type>JKS</type>
    <protocol>TLS</protocol>
    <algorithm>SunX509</algorithm>
    <authenticate-client>>false</authenticate-client>
  </keystore>
</factory>
-->
</server-sockets>
===== 以下 略 =====

```

上の内容から、まずは Keystore ファイルを作成し、そのファイル名を keystore として/conf ディレクトリ内に置けば簡便なることが分かる。後は、Keystore のパスワード部分を実際の設定内容に書き換え、コメント記号を取り除けばよい。

●Keystore の作成

Keystore の作成については、「Java チュートリアル」の「Security」を参照する(*1)。Keystore の作成は以下のようなものである（以下参照）(*2)。

```

===== Keystore の作成 =====
C:\james-2.2.0\apps\james\conf>j2sdk1.4.2_03\bin\keytool -genkey -alias james -
keyalg RSA -keystore keystore
キーストアのパスワードを入力してください: password
姓名を入力してください。
 [Unknown]: James
組織単位名を入力してください。
 [Unknown]: Gkm-Edu
組織名を入力してください。
 [Unknown]: KagawaUniv.
都市名または地域名を入力してください。
 [Unknown]: Takamatsu
州名または地方名を入力してください。
 [Unknown]: Kagawa
この単位に該当する 2 文字の国番号を入力してください。
 [Unknown]: JP
CN=James, OU=Gkm-Edu, O=KagawaUniv., L=Takamatsu, ST=Kagawa, C=JP によろ
しいですか?
 [no]: yes

<james> の鍵パスワードを入力してください。
(キーストアのパスワードと同じ場合は RETURN を押してください):

C:\james-2.2.0\apps\james\conf>dir          ← keystore の作成を確認
===== 中 略 =====
C:\james-2.2.0\apps\james\conf のディレクトリ

2004/11/14  20:27      <DIR>          .
===== 中 略 =====
2004/11/30  08:47          1,354 keystore    ←keystore が作成されている
                12 個のファイル          81,984 バイト
                3 個のディレクトリ    23,775,526,912 バイトの空き領域
C:\james-2.2.0\apps\james\conf>

```

(*1)<http://java.sun.com/j2se/1.4/ja/docs/ja/guide/security/>

(*2)上の Keystore の設定では、「姓名」に“James”を設定しているが、例えば ravie01 ホストでの設定では、“ravie01.aoki-home.kagawa-u.ac.jp”と設定する方が適切である。これについては後述する。

●起動状況等の確認

config.xml ファイルの TLS 設定に関わる部分を修正編集し、該当部分をアンコメントとして Apache James を起動する。これまで同様にエラーなく起動すればよい。

次に、config.xml ファイルのリモートマネージャに関わる設定部分で TLS 利用を“true”に設定し、4555 番ポートに telnet で接続してみる。これまでの様に ID の入力を促すメッセージが現れずハングアップ状態となれば、TLS により通信経路が暗号化されていることによるものと考えられ、設定が期待通りであることが推測される。

◆POP3S

初期設定でコメントとされている TLS の利用を有効(“true”)とし、TLS を利用したメール受信 (POP3 over SSL/TLS) を設定する。

●TLS を利用した POP3 (POP3S) の設定

TLS 利用が可能となった状況下で、config.xml ファイルの POP3 サーバの設定ブロックで <useTLS>要素の値を“true”とする (以下参照)。

```
<pop3server enabled="true">
  <port>995</port>           ← ポート番号を 110 から 995 に変更
  <!-- <bind> </bind> -->   ← コメントを外す
  <useTLS>true</useTLS>
===== 以下 略 =====
</pop3server>
```

Apache James を再起動して、次の表示が得られればよい (以下参照)。

```
C:\¥james-2.2.0¥bin>run
===== 中 略 =====
Remote Manager Service started plain:4555
POP3 Service started ssl:995           ← ssl:995 が表示されている
SMTP Service started plain:25
NNTP Service started plain:119
Fetch POP Disabled
FetchMail Disabled
```

●POP3S の検証

予め、POP3S の利用が設定された ravie01 ホストのユーザ

r-user01@ravie01.aoki-home.kagawa-u.ac.jp

宛にメールを送信しておき、このメールを end-pro-1000 ホスト上のメーラ (MUA) を用いて受信する。メーラとしては、Microsoft Outlook Express 6 と Opera 7.23 付属のものを用いる。

Outlook Express の利用

作成した r-user のアカウントとのプロパティの「詳細設定」で、次のように「このサーバはセキュリティで保護された接続(SSL)が必要」をチェックし、SSL を利用する設定を行う (図 8)。ポート番号を 995 とする。

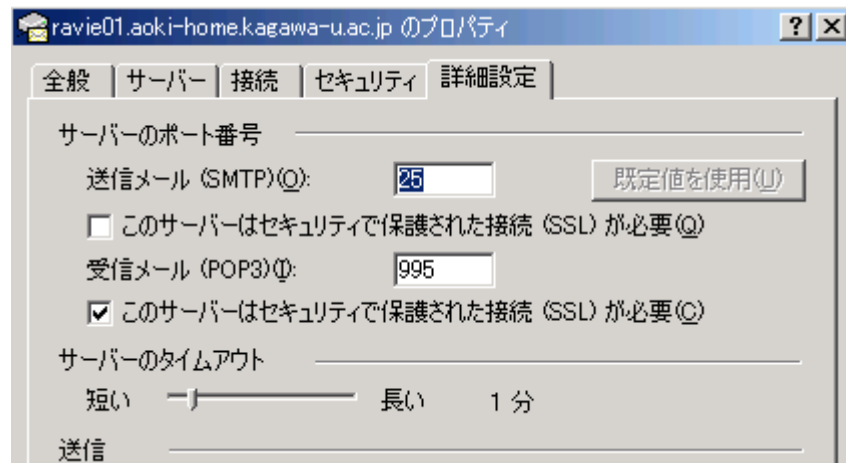


図 8 : Outlook Express での POP3S の設定

メールを受信しようとする時、通常のパスワード入力画面に続き、ravie01 ホストから送られたセキュリティ証明書が確認できないものであるが、このサーバとの接続を続けるかどうかを問うプロンプト画面が表示される (図 9)。

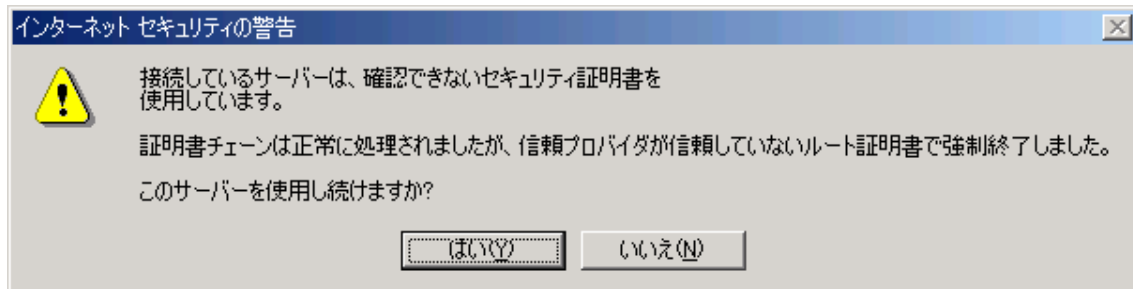


図 9 : 接続を続けるかどうかを問う画面

「はい」をクリックすると、直ちにメールが受信された(図 10)。

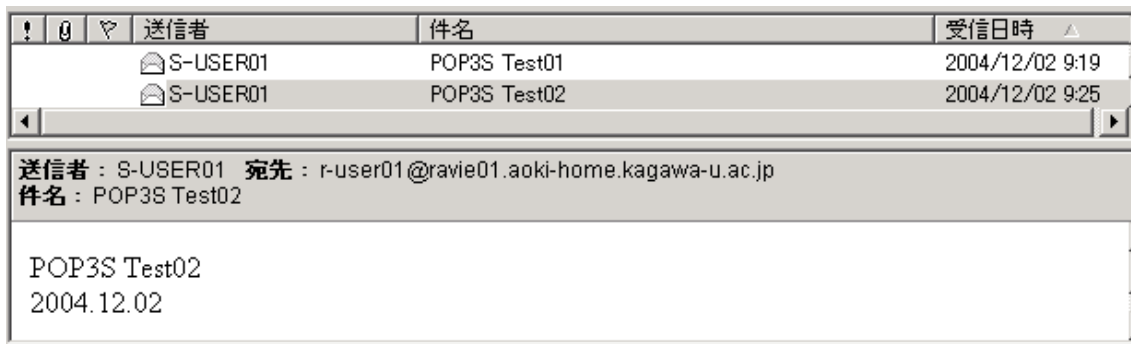


図 10 : POP3S により受信されたテストメール

Opera 7.23 付属のメーラ利用

Opera7.23 付属のメーラが Apache James の POP3S 機能とマッチしているかどうかについて検証する。SSL/TLS を利用する設定は、「メール」→「アカウントの編集」→「サーバ」の項で設定する (図 11)。



図 11 : Opera 付属メーラでの TLS 利用設定

メール受信にあたっては、先ずサーバのセキュリティ証明書を承認するかどうかの画面が表示され(図 12)、「承認する」によりパスワード入力となって、メールが受信された。

以上 2 つのメーラ(MUA)の利用により、Apache James の POP3S 機能を検証できた。

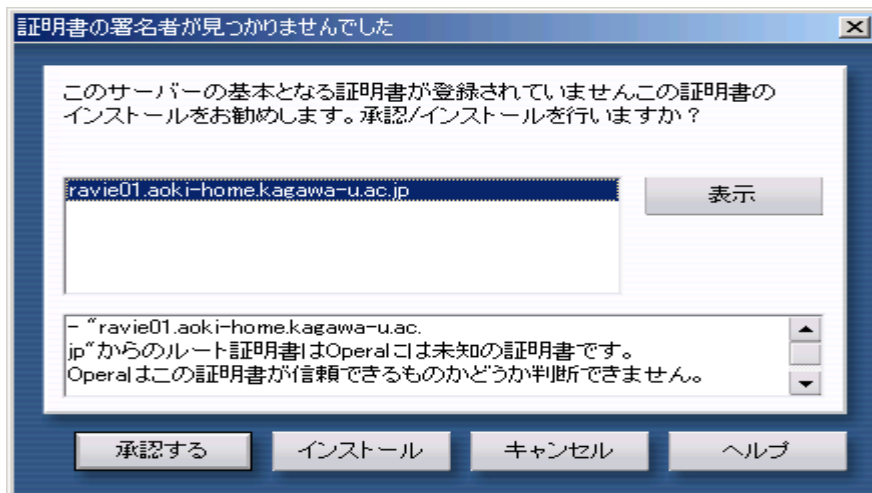


図 12 : Opera 付属メーラでの証明書承認場面

●Netscape 7.1 付属メーラと POP3S 機能

ここでは、Netscape7.1 に付属するメーラと Apache James の POP3S 機能について紹介する。結論から言えば、両者の組み合わせでは POP3S 機能はうまく動作しない。このことに関連した事柄を中心に説明する。

テストでは、上述のテストと同じく ravie01 ホストで受信されたメールを end-pro-1000 ホストから受信する。

共通名とドメイン名の不一致

POP3 接続で「保護付きの接続 (SSL) を使用する」メーラでの設定や、実際の接続に際してセキュリティ証明書の承認を求める画面等は、上記 2 つのメーラの場合とほぼ同様であるので、ここでは説明を省略する。途中、セキュリティ証明書の承認画面で証明書内容を表示させたのが図 13 である。

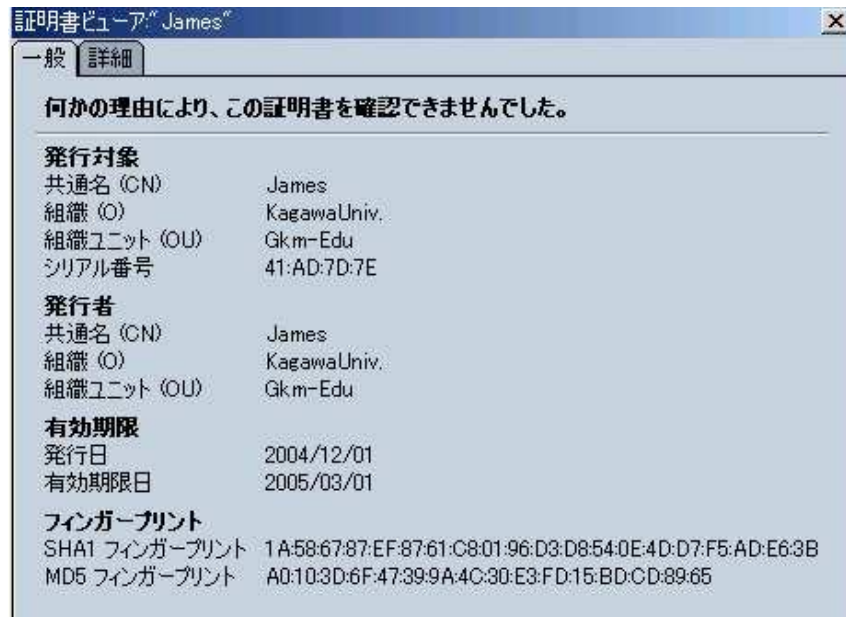


図 13 : 証明書ビューアで示された内容

図 13 の「共通名(CN)」が“James”となっている。これは先の Keystore 作成において、「姓名」として入力したものである。この共通名が”James”であることに関して次の「ドメイン名の不一致」画面が表示される (図 14)。

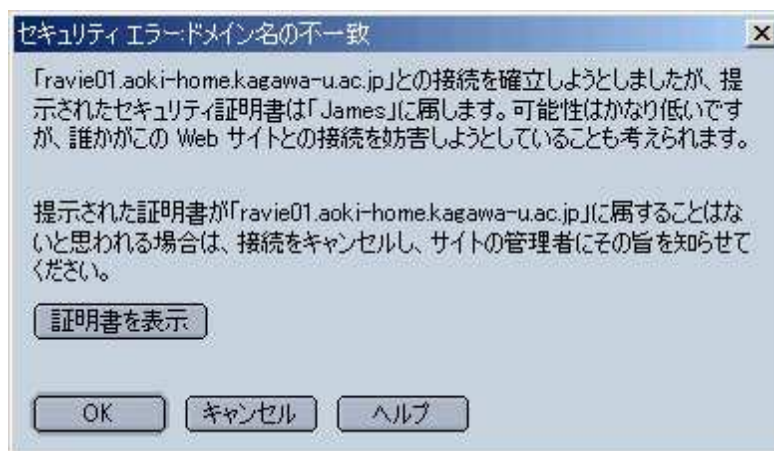


図 14 : 共通名が“James”のとき表示される「ドメイン名の不一致」

この画面は、接続先の ravie01 のドメインと”James”が一致しないために表示されるもので、Keystore 作成時に「姓名」として”ravie01.aoki-home.kagawa-u.ac.jp”を設定しておけば表示されない(実は、Outlook Express を利用してのテスト時では、修正された Keystore を用いていた)。

エラーコード 12229 による接続の失敗

図 14 の画面で「OK」をクリックすれば先に進むことができるが、ここで図 15 のエラー画面が表示され、SSL/TLS を用いた接続は失敗となる^(*)。



図 15 : エラーコード 12229 の警告表示画面

このエラーに関連した、POP3 サーバ関連のログの一部を次に示す (以下参照)。

```

===== 中 略 =====
04/12/04 09:21:33 INFO  pop3server: POP3 Service started ssl:995
04/12/04 09:32:27 INFO  pop3server: Connection from
end-pro-1000.aoki-home.kagawa-u.ac.jp (192.168.0.2)
04/12/04 09:32:32 ERROR pop3server: Exception during connection from
end-pro-1000.aoki-home.kagawa-u.ac.jp (192.168.0.2) : Connection has been shutdown:
javax.net.ssl.SSLException: Algorithm missing:
javax.net.ssl.SSLException: Connection has been shutdown:
javax.net.ssl.SSLException: Algorithm missing:
    at com.sun.net.ssl.internal.ssl.SSLSocketImpl.d(DashoA6275)
===== 中 略 =====
Caused by: java.security.NoSuchAlgorithmException: Could not create cipher AES/128
    at com.sun.net.ssl.internal.ssl.CipherBox$JCECipherBox.a(DashoA6275)
    at com.sun.net.ssl.internal.ssl.SunJSSE_h.a(DashoA6275)
    at com.sun.net.ssl.internal.ssl.CipherSuite$BulkCipher.a(DashoA6275)
    at com.sun.net.ssl.internal.ssl.SunJSSE_ax.b(DashoA6275)
    ... 18 more
Caused by: java.security.NoSuchAlgorithmException: No implementation for
AES/CBC/NoPadding found
    at com.sun.net.ssl.internal.ssl.SunJSSE_i.d(DashoA6275)
    at com.sun.net.ssl.internal.ssl.SunJSSE_i.a(DashoA6275)
    at com.sun.net.ssl.internal.ssl.CipherBox$JCECipherBox.<init>(DashoA6275)
    ... 22 more

```

この記録から、SSL ハンドシェイクがある程度進んだ段階で、恐らくはクライアントからの暗号化モード変更要求あたりの段階で、サーバ側で例外が発生したものと推測される(*2)。

以上に示したように、Netscape7.1 付属のメーラを用いての POP3S 機能の利用については、現段階では確証が得られていない。この点については残された課題としたい。

(*1)エラーコード 12229 の内容は

```

SSL_ERROR_HANDSHAKE_UNEXPECTED_ALERT-12229
"SSL peer was not expecting a handshake message it received."

```

となっている。

<http://www.mozilla-japan.org/projects/security/pki/nss/ref/ssl/sslerr.html#1040263>
を参照。

(*2)同様と思われるエラーについての報告が、次の Web ページに見られる。

<http://www.mail-archive.com/server-dev@james.apache.org/msg02780.html> :
[jira] Created: (JAMES-301) Pop3 TLS and Thunderbird 0.7 do not go together

◆SMTP 接続での TLS 利用

POP3S に続いて、SMTP over SSL/TLS 機能を検証する。

●TLS 利用の設定

TLS 利用が可能となった設定下で、config.xml ファイルの SMTP サーバの設定ブロック

で<useTLS>要素の値を”true”とする。設定はそれだけである。Apache James の起動時、次のように表示が得られればよい (以下参照)。

```
C:\¥james-2.2.0¥bin>run
===== 中 略 =====
James 2.2.0
Remote Manager Service started plain:4555
POP3 Service started ssl:995
SMTP Service started ssl:25          ← ポート 25 番で SSL 利用可能
===== 以下 略 =====
```

●メーラを用いての SMTP over SSL/TLS の検証

ravie01 ホストにおいて SMTP over SSL/TLS の設定を行った上で、end-pro-1000 ホストからメーラを用いて ravie01 ホストのユーザ宛にメールを送信する。メーラでの SMTP サーバ設定は ravie01 ホストである。簡単のため、SMTP サーバ ravie01 側での SMTP AUTH 機能は外しておく。

テスト結果は用いた 3 つのメーラ全てで、メールの送信はできなかった。理由は、次の通りである。

◆Apache James が提供する SSL/TLS 利用機能は、接続のはじめから SSL/TLS を用いた接続であること。このことは 25 番ポートに telnet で接続してみるとよく分かる。通常の “Greeting Message “が表示されず、切断した後、サーバの smtpserver に関わるログを調べると、” SSLHandshakeException “が生じたとの記録が見られるのである。

◆SMTP サーバへの接続に SSL/TLS を用いると設定したときの各メーラが提供する機能は、いわゆる STARTTLS の機能であること。STARTTLS では、通常の SSL/TLS を用いない接続で EHLO コマンドを送信した後、“250 STARTTLS”がサーバから返信されたのを受けて、SSL/TLS 接続のためのネゴシエーションに入るのである^(*)。

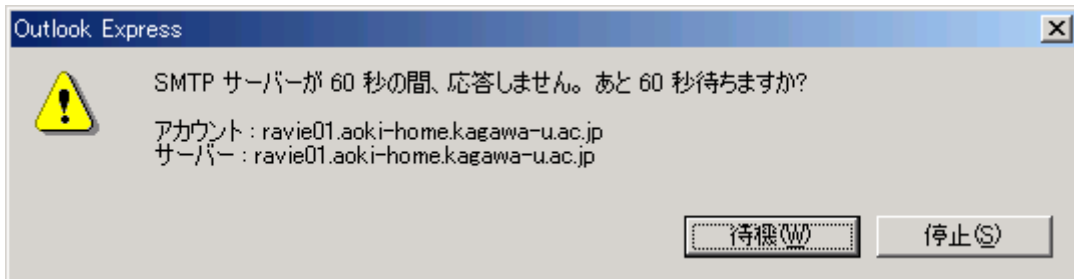


図 16 : Outlook Express でのタイムアウト表示画面

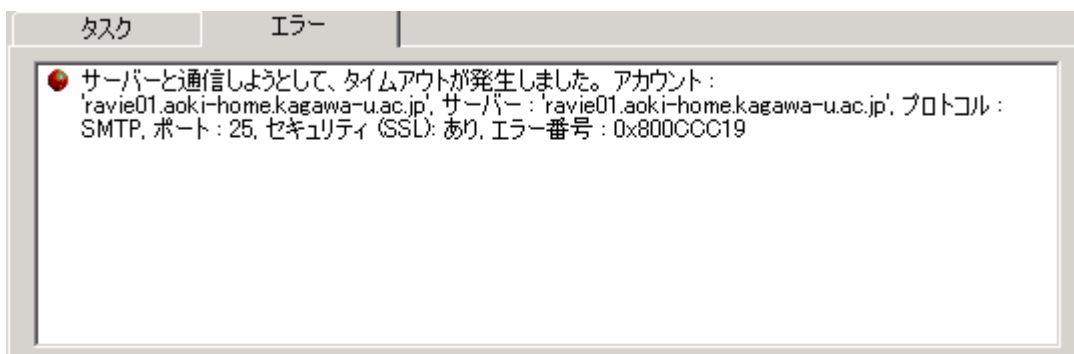


図 17 : Outlook Express でのエラー表示 (SSL 利用が示されている)

このように、接続の出発点の違いから、メーラでの SSL/TLS を用いる設定が意味をなさず、サーバ側では“SSLHandshakeException：リモートホストが途中で切断してしまった”となり、メーラ側では“timeout”となってしまう（図 16, 17）。

(*1)STARTTLS については文献(6)を参照してほしい。文献では SMTP サーバ Sendmail での STARTTLS 設定の他、各種 MUA(Mail User Agent)の STARTTLS 対応状況についても詳しく紹介されている。

●STARTTLS への期待－SMTP AUTH 機能との併用－

Apache James では SMTP AUTH 機能が提供されているが、そこで用いられる認証方式は LOGIN と PLAIN であり、パスワードがほぼ生の形で流れる (Base64) のはセキュリティ上好ましくない。各メーラの対応状況を考えるならば、是非 STARTTLS 機能が提供され、よりセキュアなメール送信が可能となることを期待したい。

■おわりに

Apache James について、SMTP サーバとしての拡張機能を中心に紹介した。文献(1)と本稿の内容を併せ読んでいただければ、Apache James を通常の SMTP サーバとして設定するにはほぼ十分であろうと思っている。

尚、NNTP サーバと FetchMail 機能の設定及び自作 Maillet の作成が課題として残されている。NNTP サーバ並びに FetchMail については、必要に応じてドキュメントを参照しつつ設定を試みてほしい。Maillet の自作が可能なところが、ある意味では Apache James のもつ楽しさでもある。これについては是非とも別途紹介したい。

参考文献

- (1)青木 昌三：「Apache James の設定と動作検証 (I)－SMTP サーバとしての基本的な設定－」 香川大学総合情報基盤センター年報 第2号 (2005年2月)
- (2)黒住 幸光：「Jakarta 活用指南 第16回 メールサーバ「Apache James を使う」」
JavaWorld 2004 July, アイ・ディ・ジー・ジャパン
- (3)青木 昌三：「SMTP サーバプログラム Sendmail の設定(II)
－SMTP AUTH 機能への対応－」
香川大学情報処理センター 年報 第11号 (2003年2月)
- (4)中村邦彦, 青木 昌三：「HP-UX での POP before SMTP」
香川大学情報処理センター 年報 第9号 (2001年2月)
- (5)青木 昌三：「メーリングリスト&ロームユーザ登録」
香川大学情報処理センター 年報 第10号 (2002年2月)
- (6)青木 昌三：「SMTP サーバプログラム Sendmail の設定(III)
－STARTTLS によるセキュアなメール送信－」
香川大学情報処理センター 年報 第10号 (2002年2月)

Apache James の設定と動作検証 (Ⅲ)

—Matcher, Mailet の作成と組み込み—

教育学部 (人間環境教育)

青木 昌三

■はじめに

別掲記事「Apache James の設定と動作検証 (Ⅰ), (Ⅱ)」において, Apache Software Foundation(ASF) の James プロジェクトによる電子メールアプリケーションサーバ Apache James の設定と動作検証について紹介した⁽¹⁾、⁽²⁾。本稿では, 自作 Matcher, Mailet プログラムの作成とそれらを Apache James に組み込む方法を解説する。取り扱っている内容の具体的な項目は次の通りである。

■テスト環境の構成 ■Matcher, Mailet プログラミングと組み込みの概要

■Matcher, Mailet プログラミングの実際と動作確認

以下では, 既に Apache James が動作し得る環境が整っているものとする。対象とするのは, James-2.2.0 である。

■テスト環境

以下で取り扱う Matcher と Mailet の組み込みについては, 文献(1), (2)で紹介したテスト環境の中で用いた Windows2000 ホスト PC 上で行っている。関係する事項をまとめておく。

テストホスト PC

IP アドレス : 192.168.0.2 OS : Windows2000(SP4)

ホスト名 : end-pro-1000 ドメイン名 (仮) : aoki-home.kagawa-u.ac.jp

Java2SDK のインストール先

インストールディレクトリ : c:\¥j2sdk1.4.2_03 以後, %JAVA_HOME%と表す。

Apache James のインストール先

インストールディレクトリ : c:\¥james-2.2.0 以後, %JAMES_HOME%と表す。

登録済みユーザ

登録済みユーザ : s-user01, r-user01, postmaster

postmaster のエイリアスとして s-user01 を設定

■Matcher, Mailet プログラミングと組み込みの概要

Apache James では, 受信したメールの処理を Matcher と Mailet のコンビネーションペアによって行うよう設計されており, これら 2 つのプログラムを自作し組み込むことができるのが, Apache James の大きな特徴である。

Matcher と Mailet のプログラミングと Apache James への組み込み方法の概略については,

http://james.apache.org/mailet_api_2_1.html

http://james.apache.org/custum_matcher_2_1.html

http://james.apache.org/custum_mailet_2_1.html

(同様の html ファイルが%JAMES_HOME%/docs/ 内に収められている。)に解説が与えられている(*1)。この章での内容は、これらの解説に基づいている。

(*1)文献(3)および次の URL ページに具体例が掲載されているので是非参照してほしい。
<http://iihito.dip.jp/docs/java/no16/james-6.html>
<http://www.ki.rim.or.jp/~kuro/Java/James/James-06-Maillet.html>

◆プログラミングの概要

Matcher と Maillet のインターフェースを定義しているのが Maillet API である。この API の詳細については

<http://james.apache.org/mailet/index.html>

を参照する(*1)。

(*1) <http://www.meisei-u.ac.jp/mirror/apache/dist/james/binaries/>で公開されている james-MailletSDK-2.2.0.zip をダウンロードし、展開して得られる /james-2.2.0/mailetsdk/javadocs/index.html から参照可能である。

●Matcher プログラミング

Matcher プログラミングでは次に示す 2 つのクラスの中から、どちらか一方を継承したサブクラスを作成する。

GenericMatcher

このクラスを継承したサブクラスを作成する場合には、オーバーライドすべきメソッドとして `getMatcherInfo()`, `init()`, `match(Mail)`, `destroy()` の 4 つがあるが、自作 Matcher の場合、`match(Mail)` メソッドを実装すればよい。

`match(Mail)` メソッドには、`org.apache.mailet.Mail` 型の Apache James が受信したメールが渡されるので、その `Mail` オブジェクトから必要な情報を得て条件にマッチするメールを選別し、条件に合ったメールの受信者アドレスのセットをリターンする。尚、条件として与えられる文字列は `getCondition()` メソッドにより得られるが、条件が与えられていない場合、このメソッドでは `null` 値が返されるようになっている。

GenericRecipientMatcher

メール受信者のフィルタリングを行う場合には、`GenericRecipientMatcher` クラスを継承する。このクラスは `GenericMatcher` クラスのサブクラスであり、これを継承したクラスを作成する場合には、`matchRecipient(MailAddress)` メソッドをプログラミングして実装する。

`matchRecipient(MailAddress)` メソッドには、`org.apache.mailet.MailAddress` 型のオブジェクトが渡されるので、受信者メールアドレスが条件にマッチするかどうかを判定し、`true` か `false` の値をリターンする。条件として与えられる受信者アドレスを得るのは、`GenericMatcher` クラスを継承する場合と同様、`getCondition()` メソッドによる。

●Maillet プログラミング

Matcher により選別されたメールを処理するのが Maillet プログラムである。Maillet プログラミングでは、雛型となる `GenericMaillet` クラスを継承したサブクラスを作成し、`init()`, `destroy()`, `getMailletInfo()`, `service(Mail)` メソッドの中、主として `service(Mail)` メソッドをプログラミングして実装する。

Maillet に初期条件として渡されるパラメータは (name,value) ペアの形となっている

が、これらは `getInitParameterNames()` や `getInitParameter(String name)` メソッドによって得ることができる。

`GenericMailet` ではメールを送信するなどのさまざまな機能が利用可能であるが、それらメソッドの詳細については、クラスについてのドキュメントを参照してほしい。

◆Matcher, Mailet 組み込みの概要

自作の `Matcher` 或いは `Mailet` プログラムを作成したならば、それらが `Apache James` にロードされるよう設定しなければならない。この設定については、従来2つの方法が提示されてきた。`Mailet` の場合を例として `custom_mailet_2_1.html` から引用する(*1)。

The Mailet must be added to James' classpath so that the Mailet can be loaded by James. There are two ways to add a custom Mailet to the classpath so that James will be able to load the Mailet. These are:

1. Download the source distribution, add a jar file containing the custom files to the lib directory of the unpacked source distribution, and build a new .sar file by following the directions here. This new .sar file will now include your custom classes.

or

2. Place a jar file containing the custom class files in the lib subdirectory of the James installation. It will also be necessary to unpack the JavaMail and James jar files from the provided .sar file and add them to this directory.

以下ではこれまでの方法についての概要を簡単に示した後、`Apache James Ver.2.2.0` で可能となっている上記と異なる方法を紹介する。

(*1)%JAMES_HOME%/docs/FAQ.html の「Where do I stick classes and jars?」には次の方法が提示されている (以下引用)。

We are largely reliant on what Avalon is doing in terms of classloading, but here are a few tips and suggestions:

Stick jars in the james/lib directory and add them to the classpath in run.bat or run.sh.

Instructions for including classes for custom mailets and matchers can be found here and here respectively.

Eventually we hope to support mailet reloading and a special lib and classes directory within the james directory that custom mailets can load from, but for now these are hopefully some useful tips.

●新しく james.sar ファイルを作成する方法

文献(3)等で紹介されている方法は上記の1. に示されている方法に相当する(*1)。具体的には、%JAMES_HOME%/apps/james.sar ファイルを適当なディレクトリにコピーして展開し、その lib ディレクトリに作成した `Matcher/Mailet` プログラムの jar ファイルを配置した後、改めてもとの james.sar に含まれていた全てのファイルとともに新しく james.sar ファイルとしてまとめ直す、というものである。その上で、新しく作成した james.sar ファイルを %JAMES_HOME%/apps/ 内に再配置して `Apache James` を再起動すれば、作成した `Matcher/Mailet` がロードされる仕組みである。具体的な手順等の詳細については、文献(3)等を参照してほしい。

(*1)上記の2. に示されている方法の設定例が

<http://www.ki.rim.or.jp/~kuro/Java/James/James-06-Mailet.html> に示されている。

●Ver.2.2.0 で可能となった方法

Faq.htmlにおいて今後の課題とされているカスタム Matcher/Maillet をロードするメカニズムが、Ver.2.2.0 では既に提供されている(*1)。次の通りである。

◆作成した Matcher/Maillet の jar ファイルは

%JAMES_HOME%/apps/james/SAR-INF/lib/ に配置

◆作成した Matcher/Maillet の class ファイルは

%JAMES_HOME%/apps/james/SAR-INF/classes/ に配置

上述の james.sar ファイルを作成し直す方法に比べて極めて簡便であることが分かる。本稿では、この方法にしたがい自作 Maillet 等を Apache James に組み込んでいくこととする。

(*1) <http://james.apache.org/changelog.html> の「New Feature」の項

- [JAMES-175] Maillet class reloading を参照のこと。[JAMES-175]のリンク先は、<http://nagoya.apache.org/jira/browse/JAMES-175> であり、次の記述が見られる。

「Backported Danny's class loading enhancements from HEAD. Custom Maillet classes can go into SAR-INF/classes; JAR files into SAR-INF/lib.」

●config.xml ファイルへの追加記述

自作の Matcher 或いは Maillet プログラムを作成して用いる場合、config.xml ファイルへの追加記述が必要である。例えば、Matcher と Maillet プログラムのパッケージ名をそれぞれ mymatcher, mymaillet とすると、config.xml ファイル内の Spool Manager ブロック、<matcherpackages>と<mailletpackages>要素内に次のように記述する（以下参照）。プロセスブロック内での利用にあたっての具体的な記述は、その都度別途記述する。

```
<mailletpackages>
  <mailletpackage>org.apache.james.transport.maillets</mailletpackage>
  <mailletpackage>mymaillet</mailletpackage>      ← 追加する
</mailletpackages>
<matcherpackages>
  <matcherpackage>org.apache.james.transport.matchers</matcherpackage>
  <matcherpackage>mymatcher</matcherpackage>     ← 追加する
</matcherpackages>
```

■Matcher, Maillet プログラミングの実際と動作確認

簡単な Matcher と Maillet のプログラムを自作し、その動作を確認する。

◆james.sar ファイルの展開先

以下では、james.sar を c:\myjames-2.2.0(以後、%MYJAMES_HOME%と表す)に展開したとして説明する。このディレクトリ配下には 3 つのディレクトリ conf, meta-inf, sar-inf があり、更に sar-inf 内に lib ディレクトリがある。

Matcher, Maillet の作成プログラムの配置場所として、それぞれ lib ディレクトリ内に mymatcher, mymaillet ディレクトリ（フォルダ）を作成しておく。

◆簡単な Maillet プログラムの作成 1

既に提供されている Maillet に、現在の時刻をメール内容として返信する ServerTime がある。この Maillet を参考としつつその日本語版 ServerTimeJp を作成し、ApacheJames に組み込んでみる。

●Jar ファイルの作成と配置

%MYJAMES_HOME%\sar-inf\lib\mymailet内に ServerTimeJp.java ファイル (付録 A 参照) を置く。このコードから作成される ServerTimeJp.class の package 名が mymailet であることに注意する。

コンパイル

ServerTimeJp.java をコンパイルする (以下参照, 表示が 2 行の場合も 1 行内に入力)。

```
C:\myjames-2.2.0\sar-inf\lib>%JAVA_HOME%\bin\javac
mymailet\ServerTimeJp.java -classpath .;james.jar;mail-1.3.1.jar;mailet_1_0.jar
C:\myjames-2.2.0\sar-inf\lib>
```

myTestMailet.jar ファイルの作成

作成された ServerTime.class を myTestMailet.jar ファイルにまとめる (以下参照)。

```
C:\myjames-2.2.0\sar-inf\lib>%JAVA_HOME%\bin\jar cvf myTestMailet.jar
mymailet
マニフェストが追加されました。
mymailet/ を追加中です。(入 = 0)(出 = 0)(0% 格納されました)
mymailet/ServerTimeJp.class を追加中です。(入 = 2071)(出 = 1033)(50% 収縮されま
した)
mymailet/ServerTimeJp.java を追加中です。(入 = 3342)(出 = 1265)(62% 収縮されま
した)
C:\myjames-2.2.0\sar-inf\lib>
```

myTestMailet.jar の配置

作成された myTestMailet.jar を %JAMES_HOME%\apps\james\SAR-INF\lib 内に配置する。

●動作検証

作成した Mailet プログラム ServerTimeJp の動作を検証する。

config.xml ファイルへの記述

上述したように, config.xml ファイル内に

```
<mailetpackage>mymailet</mailetpackage>
```

の行を追加記述する。

更に, root プロセッサ内の末尾近く, transport プロセッサに処理が移される前に ServerTimeJp による処理がなされるよう, 次の記述を付け加える。

```
<!--2004.12.27 -->
    <mailet match="RecipientIs=time@end-pro-1000.aoki-home.kagawa-u.ac.jp"
class="ServerTimeJp"/>
<!-- 2004.12.27 -->
<!-- Send remaining mails to the transport processor for either local or remote
delivery -->
    <mailet match="All" class="ToProcessor">
        <processor> transport </processor>
    </mailet>
</processor>
```

メールの送信と受信

ApacheJames の再起動後, time@end-pro-1000.aoki-home.kagawa-u.ac.jp 宛にメールを送信し, 日本語表示による時刻を内容とするメールが返信されることを確認する。送信者である s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp 宛に届いたメールの内容を次に示

す (以下参照)。期待した返信内容であることが分かる。

```
Message-ID: <25921812.1104124272171.JavaMail.Administrator@end-pro-1000>
From: time@end-pro-1000.aoki-home.kagawa-u.ac.jp
To: S-USER01 <s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp>
Subject: 現在の時刻...
In-Reply-To: <41CF9951.6090505@end-pro-1000.aoki-home.kagawa-u.ac.jp>
Mime-Version: 1.0
Content-Type: text/plain; charset=ISO-2022-JP
Content-Transfer-Encoding: 7bit
Delivered-To: s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp
```

現在の時刻 : 2004 年 12 月 27 日 (月曜) 14 時 11 分 12 秒.

◆簡単な Mailet プログラムの作成 2

送られてきたメールを受信するとともに、「休暇中である」旨のメールを送信者宛に自動返信する Mailet プログラム VacationJp を作成する。これは、古くから用いられている Vavation プログラムの Apache James 版でもある⁽⁴⁾。

●プログラムの作成と配置

プログラムの作成

VacationJp.java ファイル (付録 B 参照) を作成する。このプログラムでは、送信されてきたメールを受信するために、LocalDelivery.class を継承している。更に、返信メールの内容とするためのメッセージや、自動応答により返信するメールの有効期限を config.xml ファイルから受け取るようにしている。

コンパイルと myTestMailet.jar ファイルの作成

VacationJp.java ファイルを上述の ServerTimeJp.java と同様のディレクトリに配置した上でコンパイル、作成された VacationJp.class ファイルを ServerTimeJp.class とともに MyTestMailet.jar ファイルにまとめる。手順等は、上述した通りである。

myTestMailet.jar の配置

作成された myTestMailet.jar を %JAMES_HOME%\apps\james\SAR-INF\lib¥内に配置する。

●動作検証

config.xml ファイルへの記述

config.xml ファイル内には、既に
`<mailetpackage>mymailet</mailetpackage>`
の行が追加記述されている。

テストの度に config.xml ファイルに記述を加えるのも何かと不便なので、次のような my-mailet.xml ファイルを作成し、%JAMES_HOME%\apps\james\conf¥内に置く (以下参照)。以後はこのファイルの編集によりテストに対応できる。

```
<?xml version="1.0" encoding="Windows-31J"?>
<!-- Configuration File for Mailet Test Program -->
<!-- 2004.12.27 -->
  <mailet match="RecipientIs=time@end-pro-1000.aoki-home.kagawa-u.ac.jp"
class="ServerTimeJp"/>
  <!-- 2004.12.27 -->
  <!-- 2004.12.28 -->
```

```
<mailet match="RecipientIs=r-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp"
class="VacationJp">
<msg>正月を古里で過ごすため休暇中です。</msg>
<limitday>5</limitday> <!-- 有効期限を5日とする -->
</mailet>
<!-- 2004.12.28 -->
```

次に config.xml ファイルの先頭行近く、実体宣言部分に次の記述

```
<!ENTITY mymailetConfig SYSTEM "../conf/my-mailet.xml">
```

を追加した上で、更に mymailetConfig を参照するよう、config.xml ファイルの transport プロセッサブロック内、ローカルユーザにメールを配信する記述行の直ぐ前あたりに次の記述を追加する（以下参照）。

```
<processor name="transport">
==== 中 略 ====
    &listserverConfig;
    &mymailetConfig;          ←この行を追加記述
    <!-- Is the recipient is for a local account, deliver it locally -->
    <mailet match="RecipientIsLocal" class="LocalDelivery"/>
==== 以下 略 ====
```

この記述により、ServerTimeJp に関わる config.xml ファイルへの記述は削除して差し支えない。

メールの送信と受信

ApacheJames の再起動後、r-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp 宛にメールを送信し、r-user01 にメールが配信されるとともに、送信者である s-user01 宛に「休暇中...」をサブジェクトとするメールが届くことを確認する。s-user01 に届いたメール内容を示す（以下参照）。

```
Message-ID: <25442933.1104782704078.JavaMail.Administrator@end-pro-1000>
Date: Tue, 4 Jan 2005 05:05:04 +0900 (JST)
From: r-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp
To: S-USER01 <s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp>
Subject: 休暇中...
In-Reply-To: <41D9A54F.7060001@end-pro-1000.aoki-home.kagawa-u.ac.jp>
Mime-Version: 1.0
Content-Type: text/plain; charset=ISO-2022-JP
Content-Transfer-Encoding: 7bit
Delivered-To: s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp
```

正月を古里で過ごすため休暇中です。
 件名「VacationJp のテスト」のメールは受け取っています。
 このメールは自動応答プログラムにより送信されました。

r-user01 にメールが配信されることも確認できる（以下参照）。

```
Return-Path: <s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp>
Received: from end-pro-1000.aoki-home.kagawa-u.ac.jp ([192.168.0.2])
    by end-pro-1000 (JAMES SMTP Server 2.2.0) with SMTP ID 748
    for <r-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp>;
    Tue, 4 Jan 2005 05:04:31 +0900 (JST)
Message-ID: <41D9A54F.7060001@end-pro-1000.aoki-home.kagawa-u.ac.jp>
Date: Tue, 04 Jan 2005 05:04:31 +0900
From: S-USER01 <s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp>
==== 中 略 ====
To: r-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp
```


Subject: VacationJp のテスト
 ===== 中 略 =====
 VacationJp-Maillet
 プログラムのテスト。
 2005.01.04 朝

また、my-maillet.xml の設定で

```
<limitday>-1</limitday>
```

としたとき（有効期限は前日）、自動応答によるメールが送信されないことも確認できる。

◆簡単な Matcher プログラムの作成

提供されている“SubjectIs”や“SubjectStartWith”の改良版としての、Subject 中に指定した語を含むメールを選別する簡単な Matcher プログラムを作成するとともに、作成したプログラムを jar ファイルとしてではなく class ファイルとして配置し、自作 Matcher/Maillet プログラムの Apache James への組み込みを確認する。

●Class ファイルの作成と配置

%MYJAMES_HOME%\\$sar-inf¥lib¥mymatcher¥内に SubjectContains.java ファイル（付録 C 参照）を置く。このコードから作成される SubjectContains.class の package 名は mymatcher である。

コンパイル

SubjectContains.java をコンパイルする（以下参照、表示が 2 行の場合も 1 行内に入力）。

```
C:¥myjames-2.2.0¥sar-inf¥lib>%JAVA_HOME%¥bin¥javac  
mymatcher¥SubjectContains.java -classpath .james.jar;mail-1.3.1.jar;maillet_1_0.jar  
C:¥myjames-2.2.0¥sar-inf¥lib>
```

SubjectContains.class ファイルの配置

作成された SubjectContains.class を

```
%JAMES_HOME%¥apps¥james¥SAR-INF¥classes¥mymatcher¥  
内に配置する。
```

●動作検証

config.xml ファイルへの記述

config.xml ファイル内に、SubjectContains.class がロードされるよう、上述したように次の記述を付け加える。

```
<matcherpackages>  
  <matcherpackage>org.apache.james.transport.matchers</matcherpackage>  
  <matcherpackage>mymatcher</matcherpackage> ← 追加する  
</matcherpackages>
```

my-maillet.xml ファイルへの追加記述

作成した“SubjectContains”Matcher プログラムの動作をテストするため、Subject 中に含まれる語として“ソフトウェア”、“off”、“%”の3つを指定し、Subject 中にこれら3つの語の何れかが含まれているメールには、ヘッダ部分に

SPAM : MAY-BE-SPAM

の行を附加することとする。このためには、Maillet プログラムの動作検証で用い

た%JAMES_HOME%¥apps¥james¥conf¥my-mailet.xml ファイルに次の記述を追加すればよい (以下参照)。

```
<?xml version="1.0" encoding="Windows-31J"?>
===== 中 略 =====
<!-- 2005.01.04 Test -->
<mailet match="SubjectContains=ソフトウェア, off, %" class="AddHeader">
<name> SPAM </name>
<value> MAY-BE-SPAM </value>
</mailet>
```

これらの準備が整ったならば Apache James を再起動する。エラーを生じることなく Apache James が起動できたならば, SubjectContains.class がロードされている。

メールの送信と受信

r-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp 宛に Subject を “Hi...ソフトウェア 50% off” としてのメールを送信し, r-user01 に配信されたメールに期待通りのヘッダ行が追加記述されていることを確認する。r-user01 に届いたメール内容を示す (以下参照)。

```
Return-Path: <s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp>
===== 中 略 =====
To: r-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp
Subject: Hi...ソフトウェア 50% off ← Subject の内容
Content-Type: text/plain; charset=ISO-2022-JP
Content-Transfer-Encoding: 7bit
SPAM: MAY-BE-SPAM ← 追加されたヘッダ行
Delivered-To: r-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp
```

ソフトウェアが 50% Off で手に入ります。
次の URL へ接続を！

以上で Matcher プログラム “SubjectContains” の Apache James への組み込みとその動作が検証された。

■おわりに

自作 Matcher と Mailet プログラムの Apache James への組み込みを中心に紹介した。文献(1), (2)と本稿の内容を読んでいただければ, ほぼ Apache James を自由自在に設定できるものと思っている。Apache James を身近な SMTP サーバとして是非活用していただきたい。

参考文献

- (1)青木 昌三:「Apache James の設定と動作検証 (I) -SMTP サーバとしての基本的な設定-」 香川大学総合情報基盤センター年報 第2号 (2005年2月)
- (2)青木 昌三:「Apache James の設定と動作検証 (II) -拡張的機能の設定-」 香川大学総合情報基盤センター年報 第2号 (2005年2月)
- (3)黒住 幸光:「Jakarta 活用指南 第16回 メールサーバ「Apache James を使う」」 JavaWorld 2004 July, アイ・ディ・ジー・ジャパン
- (4)青木 昌三:「FreeBSD におけるメール関連プログラム-その設定と動作確認-」 香川大学情報処理センター 年報 第7号 (1999年2月)

〔付録〕

付録 A : ServerTimeJp.java ファイル

以下の内容は、オリジナルな ServerTime.java を参考として、日本語での時刻を内容として返信するよう書き直したものである。MailetAPI については、本文中で紹介した

<http://james.apache.org/mailet/index.html>

を参考とし、また

http://java.sun.com/j2ee/sdk_1.3/techdocs/api/javax/mail/internet/MimeMessage.html

等をも参考としている。オリジナルな Mailet プログラム等の Copyright は、そのソースコードの先頭に記されている通りである（以下参照）。

Copyright (c) 2000-2004 The Apache Software Foundation.

All rights reserved.

ServerTimeJp.java : (1行内に記すべきコードを便宜上折り返して表している)

```
package mymailet;    //パッケージ名は mymailet

import org.apache.mailet.*;
import javax.mail.*;
import javax.mail.internet.*;
import java.util.*;

public class ServerTimeJp extends GenericMailet {
    // service(Mail mail)の内容を記述する
    public void service(Mail mail) throws MessagingException {
        //返信用のメッセージを作成
        MimeMessage response = (MimeMessage)mail.getMessage().reply(false);
        response.setSubject("現在の時刻..."); // サブジェクトの内容設定
        // 時刻等を得る
        Date currentDate= new Date();
        String dateString=(new java.text.SimpleDateFormat("yyyyMMddEEEEEE"))
            .format(currentDate);
        String timeString=(new java.text.SimpleDateFormat("HH:mm:ss"))
            .format(currentDate);
        String year = dateString.substring(0,4) + "年";
        String month= dateString.substring(4,6) + "月";
        String day = dateString.substring(6,8) + "日";
        String weekday = " (" + getWeekdayJp(dateString.substring(8)) + ") ";
        String hour = timeString.substring(0,2) + "時";
        String minute = timeString.substring(3,5) + "分";
        String second = timeString.substring(6,8) + "秒";
        StringBuffer textBuffer =
            new StringBuffer(128)
                .append("現在の時刻 : ")
                .append(year)
                .append(month)
                .append(day)
                .append(weekday)
                .append(hour)
                .append(minute)
                .append(second)
                .append(".");
    }
}
```

```

String mail_body = textBuffer.toString();
// 返信メールの内容を設定, (注)を参照のこと
try{
    response.setText(mail_body,"ISO-2022-JP");
    response.setHeader("Content-Transfer-Encoding","7bit");
}catch(Exception e){response.setText(mail_body);}

// 以下は万ーのため
// From が設定されていなかったら, 届いたメールの受信者を From:に設定
if (response.getFrom() == null) {
    response.setFrom(((MailAddress)mail.getRecipients().iterator().next())
        .toInternetAddress());
}
// 返信メールの宛先が設定されていなかったならば,
// 届いたメールの送信者を宛先に設定
if (response.getAllRecipients() == null) {
    response.setRecipients(MimeMessage.RecipientType.TO,
        mail.getSender().toString());
}
response.saveChanges();
getMailletContext().sendMail(response); // メールを送信
}
/* 曜日を返す */
String getWeekdayJp(String s){
    if (s.equals("Sunday"))
        return "日曜";
    else if (s.equals("Monday"))
        return "月曜";
    else if (s.equals("Tuesday"))
        return "火曜";
    else if (s.equals("Wednesday"))
        return "水曜";
    else if (s.equals("Thursday"))
        return "木曜";
    else if (s.equals("Friday"))
        return "金曜";
    else if (s.equals("Saturday"))
        return "土曜";
    else
        return s;
}

public String getMailletInfo() {
    return "ServerTimeJp Maillet";
}
}

```

(注)返信内容の設定において,

```

response.setText(mail_body,"ISO-2022-JP");
response.setHeader("Content-Transfer-Encoding","7bit");

```

としたのが本文中で紹介した返信メールの内容である。単に

```

response.setText(mail_body);

```

とした場合, 返信されるメールのヘッダ部分で charset=Shift_JIS などとなる (以下参照)。

```

Message-ID: <1677625.1104121239390.JavaMail.Administrator@end-pro-1000>
From: time@end-pro-1000.aoki-home.kagawa-u.ac.jp
To: S-USER01 <s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp>
Subject: 現在の時刻...

```

```
In-Reply-To: <41CF8D76.4040103@end-pro-1000.aoki-home.kagawa-u.ac.jp>
Mime-Version: 1.0
Content-Type: text/plain; charset=Shift_JIS
Content-Transfer-Encoding: base64
Delivered-To: s-user01@end-pro-1000.aoki-home.kagawa-u.ac.jp
```

現在の時刻：2004年12月27日（月曜） 13時20分39秒。

付録 B : VacationJp.java ファイル

以下の内容は、オリジナルな LocalDelivery クラスを継承し、付録 A に示した ServerTimeJp.java の内容を一部引用する形で書き直している。

VacationJp.java : (1行内に記すべきコードを便宜上折り返して表している)

```
package mymailet;
import org.apache.james.transport.mailets.LocalDelivery;
import org.apache.mailet.*;
import javax.mail.*;
import javax.mail.internet.*;
import java.util.*;

public class VacationJp extends LocalDelivery {
/* Sample configuration:
 * <mailet match="RecipientIs=name@server.domain" class="VacationJp">
 *   <msg>正月を古里で過ごすため、休暇中です(optional)</msg>
 *   <limitday>5(optional,default=7)</limitday>
 * </mailet>
 */
/* ***** */
    String msg;
    String limit0;
    public void init(){
        // super.init();
        msg = getInitParameter("msg");
        if (msg == null) {
            msg="休暇中です。";
        }
        limit0 = getInitParameter("limitday");
        if (limit0 == null) {
            limit0 = "7";
        }
        limit0=getLimitDay(limit0);
    }
// service(Mail mail)の内容を記述する
    public void service(Mail mail) throws MessagingException {
//日時を得る
        String dateString=(new
java.text.SimpleDateFormat("yyyyMMddEEEEEE").format(new Date());
        int limit = Integer.parseInt(dateString.substring(0,8));
        if(limit <= Integer.parseInt(limit0)) { // 期限内ならば返信
//返信用のメッセージを作成
            MimeMessage response = (MimeMessage)mail.getMessage().reply(false);
            response.setSubject("休暇中..."); // サブジェクトの内容設定
            StringBuffer textBuffer =
                new StringBuffer(256)
                    .append(msg)
                    .append("¥r¥n")
                    .append("件名 「")
```

```

        .append(mail.getMessage().getSubject()) // サブジェクトを取得
        .append("」 の")
        .append("メールは受け取っています。")
        .append("¥r¥n")
        .append("このメールは自動応答プログラムにより送信されました。")
        .append("¥r¥n");
    String mail_body = textBuffer.toString();
    // 返信メールの内容を設定
    try{
        response.setText(mail_body,"ISO-2022-JP");
        response.setHeader("Content-Transfer-Encoding","7bit");
        response.setSentDate(new Date());
    }catch(Exception e){response.setText(mail_body);}
// 以下は万一のため
// From が設定されていなかったら、届いたメールの受信者を From:に設定
    if (response.getFrom() == null) {
response.setFrom(((MailAddress)mail.getRecipients().iterator().next()).toInternetAddress());
    }
// 返信メールの宛先が設定されていなかったならば、
// 届いたメールの送信者を宛先に設定
    if (response.getAllRecipients() == null) {
        response.setRecipients(MimeMessage.RecipientType.TO,
mail.getSender().toString());
    }
    response.saveChanges();
    getMiletContext().sendMail(response); // メールを送信
} // メール返信,終わり
}
/* 有効期限を返す */
String getLimitDay(String s){
    int day;
    try{
        day = Integer.parseInt(s);
    }catch(Exception e){ day = 7;}
    Calendar cal = Calendar.getInstance();
    cal.add(Calendar.DATE,day);
    String dateString=(new
java.text.SimpleDateFormat("yyyyMMddEEEEEE").format(cal.getTime());
    return dateString.substring(0,8);
}
/* 有効期限を返す, おわり */
public String getMiletInfo() {
    return "VacationJp Milet";
}
}

```

付録 C : SubjectContains.java ファイル

SubjectContains.java :

```

package mymatcher;
import org.apache.milet.*;
import javax.mail.internet.*;
import java.util.*;
public class SubjectContains extends GenericMatcher {
    public Collection match(Mail mail) throws javax.mail.MessagingException {
        String subject = mail.getMessage().getSubject();
        StringTokenizer st = new StringTokenizer(getCondition(), " ¥t", false);
    }
}

```

```
int n=0 ;
if(subject != null){
    while (st.hasMoreTokens() {
        if(subject.indexOf(st.nextToken()) != -1)
            ++n ;
        }
    }
if (n >= 1)
    return mail.getRecipients();
else
    return null;
}
}
```

教育用 PC システムの保守・更新作業について

教育学部 黒田 勉
工学部 香川 考司
幸町分室 曽根 計俊, 瀬野 芳孝

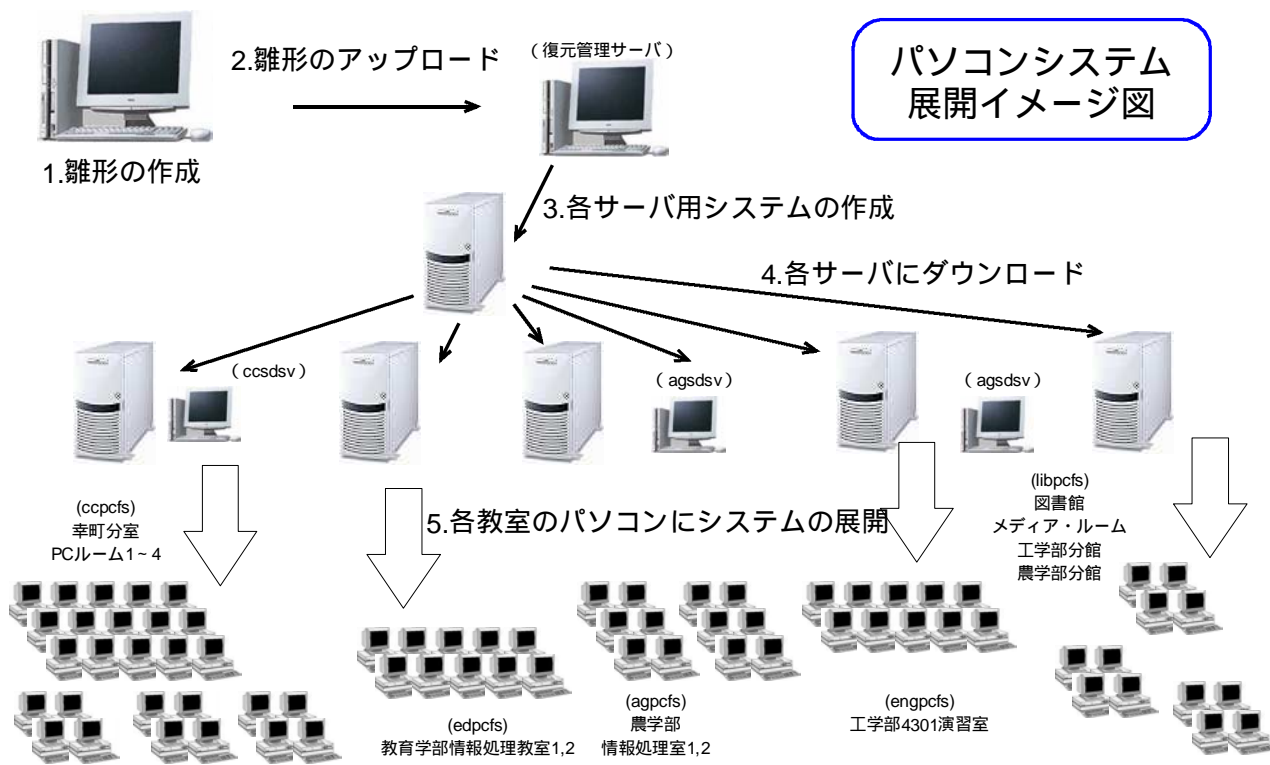
旧香川大学情報処理センターから引き継いだ教育用 PC のシステム保守・更新作業は、夏休中、年末年始、春休み中の年3回を定例とし、システムに深刻なバグが報告された時には随時、下記の工程表に従って行われている。

工程表並びに展開イメージ

1. 学生利用パソコンにて雛形の作成 約1週間から10日間必要

OS(WindowsXP)を最新版に変更(セキュリティパッチの適用等)

アプリケーションを最新版に変更および動作確認(これが非常に面倒:チェック者...黒田・香川(全般)、水野(英語学習システム)、学務課担当者(履修登録)、曽根、必要に応じて導入ソフト担当者)



2. 雛形をサーバにアップロード

3. 各サーバ用システムの作成 約2日間必要

(7システム分：センター、教育学部、工学部、農学部、図書館、工学部分館、農学部分館)
本体部分を必要サーバ分作成し、サーバごとの設定情報を追加(変更)して個々のサーバ用システムを作成

4. 各サーバにダウンロード(基盤センター、教育学部、農学部、図書館、工学部分館、農学部分館) 延べ約4日間程度必要

5. 各サーバよりクライアント PC に展開(教室に設置の台数分) 1教室に2時間程度

雛形作成に時間を要する理由は下記の通りである。

- ・ チェックそのものに時間を要する。
起動だけであれば終わりではなく、学生がやりそうなことを一通り試しておく必要がある。
- ・ システムに依存関係があるため、チェックの日程調整が必要である。
担当者の出張等の都合でチェック者の日程調整が巧くいかない等。
- ・ 使用者(特に教員側)からの要望がまとまって入らない。
- ・ 要望のレベルが異なる(ある人の要望と別の人の要望が逆の場合など)。
- ・ 導入されているソフトの不具合が修正できていないことがある。
- ・ フリーソフトの場合、担当者が知らない間に修正されていることがある。
導入したソフト担当者が更新を見落としした等

昨年(2004年)は更新を1月,3月,5月,9月,10月,12月に行っている。特に,危険度の高い Windows や Office のセキュリティパッチを反映させるための更新が多くなっている。

ソフトの使用方法の変更は,動作チェック並びに対策に時間的余裕のとれる7月末から8月に入念に行うことが通例となっている。これは,新学年開始後,前期の講義の間に,ユーザーである学生や,教員の意見を拾い上げるための十分な時間をとることができることも大きい。

各教室でのクライアント PC へのダウンロードは,2時間程度で終わるが,前後の作業(動作確認のためのディスプレイ電源投入作業や,使用中の学生への周知など)に時間をとられるため,実作業時間は約3時間掛かっている。教育学部では情報処理教室を閉鎖せずに更新を行うため,2つのシステム更新に6時間弱必要である。

幸町分室では,PC ルームを締切扱いにし,別室での利用を促しているため,このような問題は発生していないが,PC ルーム 1,2,PC ルーム 3,PC ルーム 4 を同時に閉鎖で

きないため、延べ6～8時間必要である。

農学部では、2部屋が完全に分かれており、どちらかを閉鎖しても問題は殆ど生じない。1部屋ずつ締切扱いにして更新を行っている。2部屋の更新には延べ6～7時間必要である。

工学部では、学生の立ち入らない夜間に一斉に行っているため、総時間数は不明であるが、他の部局と同様、延べ7～8時間は掛かるものと考えられる。

教育学部 PC ルーム(421 教室)で行っている更新作業は、概ね下記の通りである。

1. 基盤センター幸町分室からサーバへのアップロード完了の連絡がくる。
2. 教育学部スタッフで PC ルーム並びにスタッフの空き時間の調整をとる。
- ・ 1と2の順序は逆の場合もあり、空き時間までにアップロードを済ますこともある。
3. 教育学部の PC ルームは 21 台と 41 台がセグメント分けされているため、1 セグメントずつ更新を行う必要がある。しかも、24 時間開放を行っているため、PC を使用中の学生がおり、彼らの利便を図りながら更新をしなくてはならない。そこで、更新しない方のセグメントの PC へ移動させた後、更新作業を開始する。
4. 更新作業そのものは、NEC から提供されている「シナリオファイル」に登録した後、このシナリオを実行するだけで良いが、最初の PC 起動タイミングだけは目視して、起動できないものは、手動で起動する。毎回の作業において数台が起動しない。この確認を怠ると、起動できなかった PC はシステムが更新されず、再度更新作業をしなくてはならなくなり、二度手間になってしまう。
5. 約 2 時間 30 分で 1 セグメントの更新が完了する。完了後は、数台の PC を立ち上げ、うまく更新できたかどうかを確認する。
6. 残りのセグメントを更新するため、学生を再度移動させ、シナリオファイルを実行する。
7. 約 2 時間 30 分後、更新が完了するので、同様に更新完了のチェックを行う。
8. ちなみに、HDD 不良やマザーボード障害などの大きな PC の不具合が発見されるのはこのチェックの時が多い。

12 時頃から更新作業を始め、チェックを含めて全ての作業が終了するのは概ね 19 時である。

以上、教育用 PC の更新作業についての流れを示した。何かの役に立てば幸いである。

ウェブページアクセスログ解析

所属： 経済学部経済学科

氏名： 中村邦彦

1. はじめに

手元の記録によれば、香川大学の公式ウェブサーバーwww.kagawa-u.ac.jp が設置されたのは、1996年4月であるから、まもなく10年になる。筆者はそのサーバーが公式サーバーになる前からサーバー管理に係わってきた。当然、どのページがどのくらい閲覧されているのか、気にはなっていた。いずれ本格的にアクセスログの分析をしなければならないとは考えていたが、ここにきてようやくやってみることにした。まず、どんなツールがあるか調べたところ、最近は色々なツールやサービスが存在する。分類すると次のようになる。

・サーバーログ解析型

これはウェブサーバーのアクセスログを解析するタイプである。どのページが何回アクセスされたかということだけでなく、詳しく分析すれば、さまざまな情報が得られる。

・ページ埋め込み型

解析したいウェブページにプログラムや画像を埋め込んで、記録するタイプである。欲しい情報が得られる点はよいが、各ページに何かを埋め込まなければならないのが難点である。

・パケットモニター型

ネットワーク上のどこかで、通過するパケットをモニターするタイプである。本学のように、ウェブページが多数のサーバーに分散している場合は、有効な方法である。例えば、大学トップページに接続した利用者が、次にどのサイトへ行ったのかを追跡できる。

今回は手軽に利用できるサーバーログ解析型のツールの中から、フリーな **Webalizer** と **Analog** を使ってみることにした。まず、筆者の研究室のサーバーで試したあと、大学の公式サーバーに導入した。研究室サーバーは **Linux/Debian**、大学のサーバーは **Red Hat Linux 7.2** である。今回は、ツールの機能と使い方を調べるのが目的であり、アクセス解析そのものが主目的ではないので、その点はご了承ください。

2. Webalizer

Webalizer は **Bradford L. Barrett** が開発し、**GPL** に従って配布されているウェブアクセスログ解析ツールである。次のような統計結果がグラフや表で出力される。もうひとつのツール **Analog** に比べて、見栄えがよいので、まずこちらから紹介する。

- 月の統計
- 日ごとの統計
- 時間ごとの統計
- ヒット数ランキング/URL 別、転送量別、Entry Pages 別、Exit Pages 別、サイト別、転送量
- リファラー(どのページからのやってきたか)
- 検索文字列(Yahoo 等の検索エンジンからどういう検索文字で検索されたか)
- ユーザーエージェント(ブラウザ、OS)
- 国別統計

2.1 Webalizer のインストール

Webalizer は、Debian と Red Hat にはパッケージの形で含まれているので、導入はいたって簡単である。以下はDebian Sarge の場合である。Webalizer の最新バージョンは2.01-10であり、Debian Sarge では2.01.10-26となっている。

"apt-get install webalizer" でインストールすると、次のように設定項目を聞いてくるので、適宜応えれば、とりあえずインストールは完了する。

```
Configuring webalizer
Which directory should webalizer put the output in?
/var/www/webalizer
<Ok>
```

図1 Webalizer の設定ダイアログボックス 1

```
Configuring webalizer
(your system's hostname will be appended to it)
Enter the title of the reports webalizer will generate.
Usage Statistics for is.ec.kagawa-u.ac.jp
<Ok>
```

図2 Webalizer の設定ダイアログボックス 2

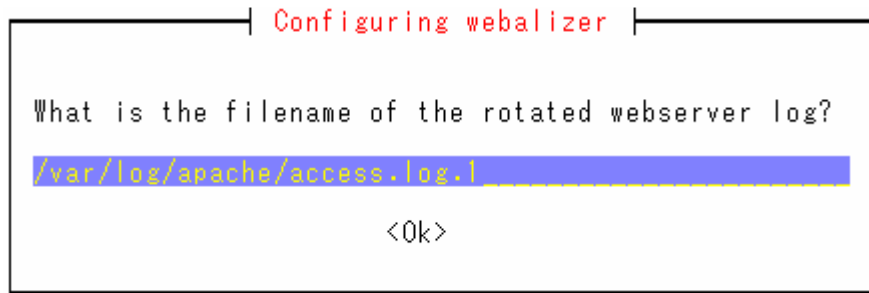


図3 Webalizer の設定ダイアログボックス 3

アクセスログファイルは、そのシステムに合わせて設定する。注意する点としては、webalizer を cron で毎日実行する場合、ログファイルがローテイトされるタイミングで問題が起こらないようにしなければならない点である。Debian の場合は、ここでは access.log ではなく、access.log.1 がデフォルトになっている。

webalizer の実行プログラム本体は /usr/bin/webalizer であるが、/usr/bin/webalizer から webalizer へシンボリックリンクが張られている。

```
lrwxrwxrwx 1 root root 9 2005-01-28 09:57 webalizer -> webalizer
```

設定ファイルは /etc/webalizer.conf、webalizer が出力するウェブページは /var/www/webalizer になる。その他、各国語に対応したメッセージが用意されていて /usr/share/locale/ の下に置かれる。webalizer を毎日1回起動するためのスクリプト /etc/cron.daily/webalizer が用意されている。

以上は Debian の場合で、Red Hat では少々違いがあるが、ここでは説明を省略する。

この段階で webalizer を起動すれば、何らかの結果が得られるはずである。このとき、環境変数 LANG を日本語の設定にしていると、出力されるメッセージが日本語になるのだが、おそらく画像中の文字が正しく日本語にならず、文字化けしてしまうであろう。これは、グラフィックライブラリーGD が正しく日本語フォントを使えるようになっていないためである。今回はその点は目をつむることにし、webalizer を実行するときは、LANG を C に設定しておくことにする。

2.2 Webalizer の用語

ここで Webalizer の出力を理解するために、用語を簡単に説明しておく。単にアクセス数といってもいろいろな数え方が考えられるからである。後に Analog の出力例も掲載しているが、Analog は別の言い方を採用している。

Hits(ヒット数)

サーバーに対して行われたあらゆるリクエストの数がヒットである。エラーも含まれる。ただし、設定ファイルで除外するように指定したものは除かれる。

Files(ファイル数)

リクエストしたクライアントに対して、何らかのレスポンスを正常に返したものの。

Pages(ページ数)

Webalizer README には、“Pages are, well, pages!” と書いてあるように、定義しにくいですが、HTML ファイル等がこれにあたる。画像や CSS ファイル等、あるページに付属するものは含まない。何をページとして数えるかは、設定ファイルで指定することができる。

Kbytes(バイト数)

サーバーから送信されたデータのバイト数

Sites(サイト数)

IP アドレスで識別される接続クライアントの数。集計期間内では、同じアドレスから何回アクセスがあっても1サイトと数える。

Visits(訪問者数)

同一のアドレスからのアクセスであっても、直前のアクセスから、設定ファイルで指定された visit timeout の値より長く経過している場合は新しい訪問として数えられる。

2.3 検索文字の文字化け対策

サーバー内のページにアクセスしてくる人が、そのページを、どこかの検索サービスを利用して見つけている場合、そのときの検索文字が Referer¹フィールドに含まれている。ログ解析プログラムではこの検索文字の統計をとってくれるのであるが、この検索文字は日本語であることが多いので、URL の中では%hh のように16進数でエンコードされている。したがってそのまま表示したのではわかりにくい。そこで、この検索文字をできるだけ正しく表示するように工夫した。

まず、ウェブを検索したところ、次のような Perl スクリプトが見つかった²。

```
#!/usr/bin/perl
use Jcode;
while(<>){
    $_ =~ s/(%|¥¥x) ([a-fA-F0-9] [a-fA-F0-9])/pack("C", hex($2))/eg;
    Jcode::convert(¥$_, euc);
    print;
}
```

これで%と16進数2桁で表されている日本語文字を、もとの文字コードに変換できるのであるが、中には先頭の%自身を%25としてエンコードしているものがあるため、1度に変換することはできない。ま

¹ 参照元という英語は Referrer が正しいのだが、RFC2616には Referer と定義されている。ウェブの初期に間違えたものだが、そのまま使われている。本稿では適宜使い分けた。

² <http://acorn.zive.net/~oyaji/www/webalizer.htm>

た、最近ではマルチバイト文字を UTF-8 でエンコードしているケースが増加しているため、出力を UTF-8 にした。以下に、実際に使用している Perl スクリプトを示す。

```
#!/usr/bin/perl

use Jcode;
$logfile = shift @ARGV;

if (defined($logfile)) {
    open (IN, "<$logfile");
    $fh = ¥*IN;
} else {
    $fh = ¥*STDIN;
}
while(<$fh>){
    $_ =~ s/%25/%/g;
    $_ =~ s/(%|¥¥x) ([a-fA-F0-9] [a-fA-F0-9])/pack("C", hex($2))/eg;
    Jcode::convert(¥$_, utf8);
    print;
}
# close(IN);
```

これでかなりの文字列が正しく表示されるようになったのだが、中にはまだ文字化けしているものがあった。原因を調べてみると、一行の中に異なる文字コードが含まれている場合と、文字コードを変換している Perl モジュール Jcode.pm が、文字コードを間違って判断している場合であった。

HTML の出力を UTF-8 にしたので、その中の META タグを次のように指定したい。

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

それで、`/etc/webalizer.conf` に次の設定を追加した。

```
HTMLHead <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
```

ところが、これで実行したところ、この META 行が2行表示される。マニュアルを確認したら、次のように書いてある。

コマンドラインのオプション `-c` について

`-c file` Use configuration file `file`.

A default configuration file is scanned for. A file named `webalizer.conf` is searched for in the current directory, and if found, its configuration data is parsed. If the file is not present in the current directory, the file `/etc/webalizer.conf` is searched for and, if found, is used instead.

これだけ読んで、`-c` で設定ファイルを指定したら、`default` の設定ファイルは使わないものと考えたのだが、実際には両方の設定ファイルが処理されていた。

なお、設定ファイル `webalizer.conf` の中で、次のように検索文字の設定をしておかないと、かなりの数の検索文字を見逃すことになるらしい。³ ただし、米国の検索エンジンの情報は内蔵されているので、`webalizer.conf` の中では設定しなくてもよい。

```

SearchEngine    google.co.jp    q=
SearchEngine    yahoo.co.jp    p=
SearchEngine    search.biglobe.ne.jp    q=
SearchEngine    infoseek.co.jp    qt=
SearchEngine    search.nifty.com    Text=
SearchEngine    goo.ne.jp      MT=
SearchEngine    excite.co.jp   search=
    
```

2.4 香川大学公式 WWW サーバーの解析結果

以下、香川大学の公式 WWW サーバーのアクセスログを `Webalizer` で処理した結果を簡単に紹介する⁴。

トップページには過去1年間の月別統計のグラフと表が表示される。表のうち、月名が各月別分析のページへのリンクになっている。

`Hits` と `Files` に比べて、`Pages` が非常に少なくなっている。実際、`Pages` は `Files` の9%である。経済学部のサーバーの場合は60%であるので、これに比べればかなり小さいことがわかる。これは、おそらく、トップページの作り方のせいである。このページは50以上のファイルから構成されており、ページ1に対してファイルが50にもなる。そして、そのページへのアクセスが最も多いのである。

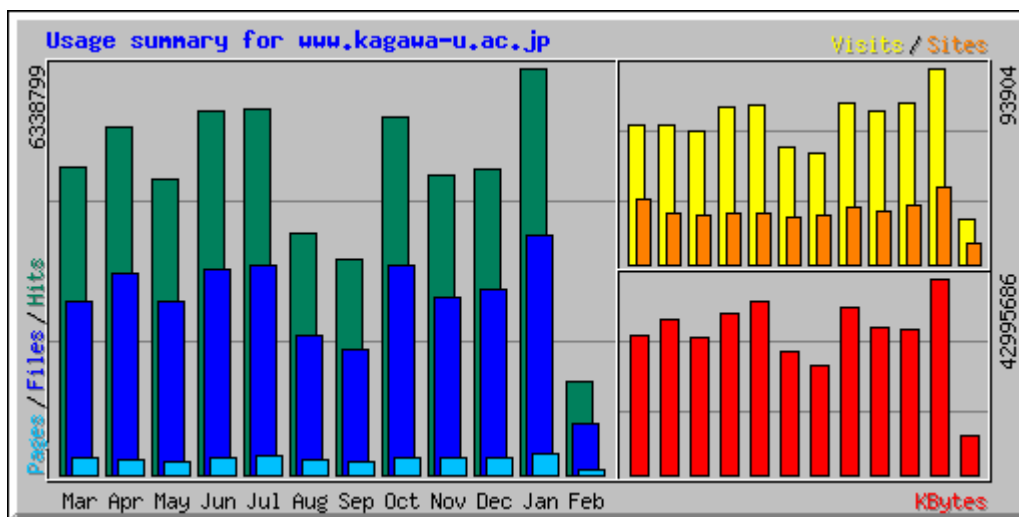


図4 Usage summary for www.kagawa-u.ac.jp(2005/2/6)

³ `webalizer` の検索語の日本語対応の話 <http://tyche.pu-toyama.ac.jp/~a-urasim/webalizer/>

⁴ 解析結果はウェブブラウザで表示できるが、一部に公開すべきかどうか判断に迷うものが含まれているため、現在は公開していない。

Summary by Month										
Month	Daily Avg				Monthly Totals					
	Hits	Files	Pages	Visits	Sites	KBytes	Visits	Pages	Files	Hits
Feb 2005	239552	134077	11901	3614	10043	8686747	21688	71410	804467	1437315
Jan 2005	204477	120223	10660	3029	36519	42995686	93904	330477	3726941	6338799
Dec 2004	154046	93520	8262	2490	27795	31798652	77201	256150	2899146	4775452
Nov 2004	155965	92525	8471	2440	25023	32115966	73205	254145	2775772	4678972
Oct 2004	179677	105057	8665	2475	27341	36776209	76753	268639	3256791	5570010
Sep 2004	111819	65224	6471	1765	23194	23736274	52950	194139	1956741	3354586
Aug 2004	121495	69925	7800	1808	22641	26971995	56077	241828	2167689	3766352
Jul 2004	184162	105415	9807	2472	24133	37932985	76633	304023	3267880	5709039
Jun 2004	189423	106956	9087	2520	24592	35203665	75622	272633	3208698	5682707
May 2004	149065	86900	6592	2061	23314	30215682	63916	204358	2693903	4621029
Apr 2004	180210	104631	7803	2215	24238	34039072	66476	234096	3138938	5406300
Mar 2004	154291	87004	8738	2159	30744	30658131	66957	270884	2697136	4783044
Totals						371131064	801382	2902782	32594102	56123605

図5 Summary by Month(2005/2/6)

次に月別の結果を紹介する。これは2004年12月の集計結果である。とくに断らない限り、結果はひと続きのページに表示されている。

まず、一番上に次のような行が表示されるが、これが各項目へのリンクになっている。

[\[Daily Statistics\]](#) [\[Hourly Statistics\]](#) [\[URLs\]](#) [\[Entry\]](#) [\[Exit\]](#) [\[Sites\]](#) [\[Referrers\]](#) [\[Search\]](#) [\[Agents\]](#) [\[Countries\]](#)

Monthly Statistics for December 2004		
Total Hits	4775452	
Total Files	2899146	
Total Pages	256150	
Total Visits	77201	
Total KBytes	31798652	
Total Unique Sites	27795	
Total Unique URLs	2969	
Total Unique Referrers	2230	
Total Unique User Agents	325	
	Avg	Max
Hits per Hour	6418	56570
Hits per Day	154046	392091
Files per Day	93520	212344
Pages per Day	8262	25032
Visits per Day	2490	4818
KBytes per Day	1025763	2292824

図6 Monthly Statistics for December 2004

Hits by Response Code	
Undefined response code	8
Code 200 - OK	2899146
Code 206 - Partial Content	26205
Code 301 - Moved Permanently	790
Code 304 - Not Modified	1813494
Code 400 - Bad Request	11
Code 403 - Forbidden	1147
Code 404 - Not Found	34370
Code 405 - Method Not Allowed	40
Code 408 - Request Timeout	240
Code 501 - Not Implemented	1

図7 Hits by Response Code

Daily Statistics

他の月と違って、12月は8日と10日のアクセスが多くなっている。他の月では平日と休日の差はあるが、平日間では大きな違いはない。また12月は冬休みのため、20日以降にアクセスの減少が見られる。

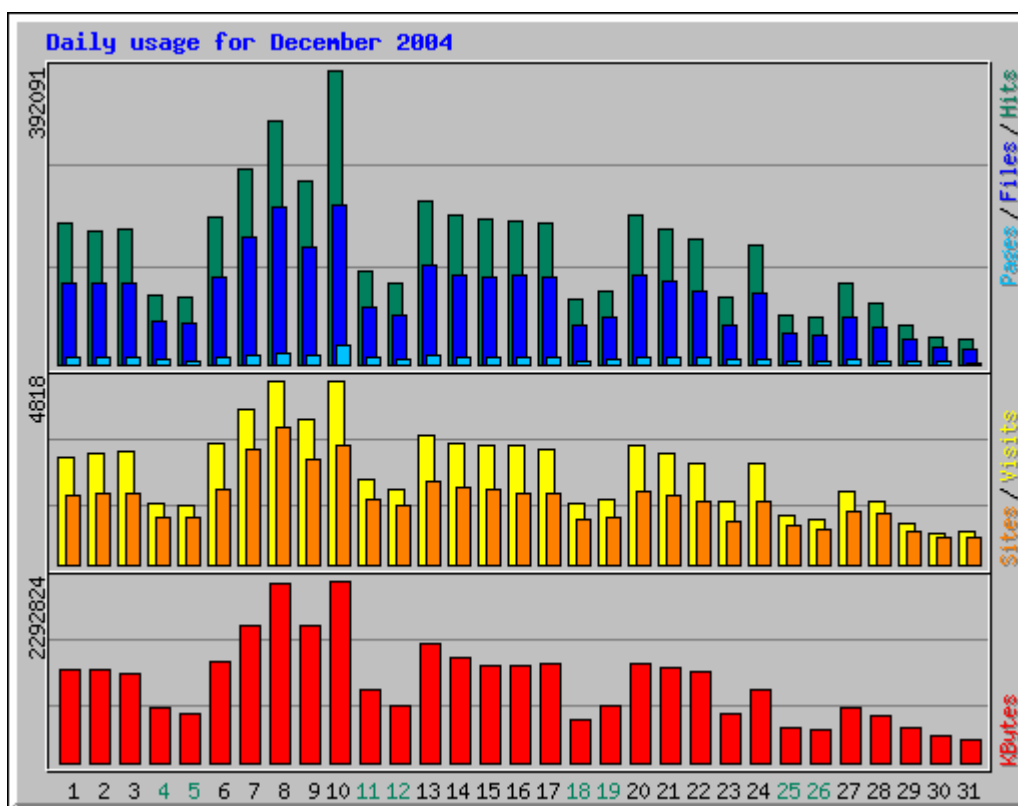


図8 Daily Usage for December 2004

Hourly Statistics

時間ごとのアクセス数はほぼ予想通りであろう。実は、毎日深夜に筆者の運用する検索ロボットがアクセスしているのだが、その分は取り除いてある。

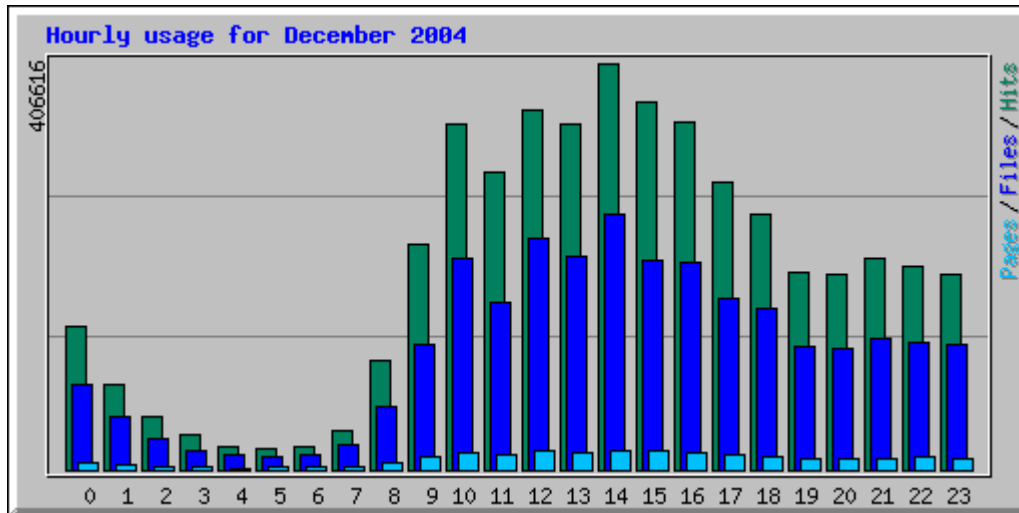


図9 Hourly Usage for December 2004

URLs

最もリクエストが多いのがトップページである(図10)。2位がサーチエンジンのリンク集であるのは意外であった。実はこのページはかなり以前に筆者が作成したもので、その後ページデザインがすっかり変更されたにもかかわらず、まだ当時のままで使われていたとは知らなかった。最近新しい検索サービスが色々現れているので、最新のものに更新して欲しい。

3位は入試のページになっている。これは予想されるであろう。表の最後に、"View All URLs"へのリンクがある。これにより、すべてのURLへのアクセス数を見ることができるが、ここでは省略した。このファイルは、デフォルトでは作成されないので、`webalizer.conf`の中で、次のような行を追加する。

```
AllURLs          yes
```

Entry

Entry ページのトップ10が表示されている(図11)。Entry ページというのは、内部のページのリンクをたどってアクセスされたのではなく、外部から直接リクエストされたページである。

`/kagawa_uni/sub1.htm` は「受験生の方」のためのページ、`% /kagawa_uni/sub4.htm` は「在学生の方」のためのページである。

Top 30 of 2969 Total URLs					
#	Hits		KBytes		URL
1	85201	1.78%	2568214	8.08%	/
2	15444	0.32%	27417	0.09%	/searchengine.html
3	12639	0.26%	78005	0.25%	/univ/nyushi/
4	10796	0.23%	115034	0.36%	/kagawa uni/lmk.htm
5	6296	0.13%	79282	0.25%	/kagawa uni/sub1.htm
6	4881	0.10%	96302	0.30%	/kagawa uni/inf.htm
7	4664	0.10%	3928	0.01%	/univ/nyushi/gakubu-m2.htm
8	4463	0.09%	1712	0.01%	/univ/nyushi/gakubu.htm
9	4440	0.09%	46756	0.15%	/kagawa uni/sub4.htm
10	4419	0.09%	81479	0.26%	/univ/nyushi/seikyul.htm
11	4414	0.09%	7569	0.02%	/univ/nyushi/gakubu-m.htm
12	4402	0.09%	95331	0.30%	/kagawa uni/hot.htm
13	4359	0.09%	6563	0.02%	/univ/nyushi/shin.htm
14	3022	0.06%	56397	0.18%	/kagawa uni/acs.htm
15	2632	0.06%	1317	0.00%	/univ/nyushi/zisshi2.htm
16	2528	0.06%	1047	0.00%	/univ/nyushi/zisshi.htm
17	2478	0.06%	2323	0.01%	/univ/nyushi/zisshi1.htm
18	2102	0.04%	20844	0.07%	/univ/nyushi/zisshi16.pdf
19	2072	0.04%	3584	0.01%	/univ/nyushi/goukaku-suisen.htm
20	1964	0.04%	20942	0.07%	/kagawa uni/msg.htm
21	1809	0.04%	14762	0.06%	/univ/nyushi/12-101.pdf
22	1801	0.04%	18337	0.06%	/univ/nyushi/suisen16.pdf
23	1686	0.04%	17707	0.06%	/univ/nyushi/17lkyouka.pdf
24	1611	0.03%	51525	0.16%	/kagawa uni/smp.htm
25	1595	0.03%	11283	0.04%	/univ/nyushi/11-8.pdf
26	1501	0.03%	24744	0.08%	/kagawa uni/kou.htm
27	1249	0.03%	12778	0.04%	/univ/nyushi/17mkyouka.pdf
28	1241	0.03%	18574	0.06%	/kagawa uni/act1.htm
29	1211	0.03%	8881	0.03%	/univ/nyushi/sigan sui 1 h14.htm
30	1174	0.02%	10596	0.03%	/kagawa uni/act2.htm

[View All URLs](#)

図10 Top 30 of URLs

Top 10 of 742 Total Entry Pages					
#	Hits		Visits		URL
1	85201	1.78%	56658	76.54%	/
2	12639	0.26%	1842	2.49%	/univ/nyushi/
3	15444	0.32%	1572	2.12%	/searchengine.html
4	4402	0.09%	929	1.25%	/kagawa uni/hot.htm
5	6296	0.13%	358	0.48%	/kagawa uni/sub1.htm
6	1018	0.02%	343	0.46%	/ccrd/
7	4419	0.09%	322	0.43%	/univ/nyushi/seikyul.htm
8	10796	0.23%	310	0.42%	/kagawa uni/lmk.htm
9	1172	0.02%	308	0.42%	/lifelong/
10	4440	0.09%	263	0.36%	/kagawa uni/sub4.htm

図11 Top 10 of Entry Pages

Exit

Exit ページは、そのページから訪問者が出て行ったページである。このサーバーのログだけを調べていたのでは、そのページから学内の他のサーバーのページに飛んで行った場合も、出て行ったとしか見えないので、ここに数えられてしまう。

Top 10 of 838 Total Exit Pages					
#	Hits		Visits		URL
1	85201	1.78%	34626	46.82%	/
2	15444	0.32%	8517	11.52%	/searchengine.html
3	4440	0.09%	2023	2.74%	/kagawa uni/sub4.htm
4	4402	0.09%	1655	2.24%	/kagawa uni/hot.htm
5	12639	0.26%	1622	2.19%	/univ/nyushi/
6	10796	0.23%	1568	2.12%	/kagawa uni/lmk.htm
7	4419	0.09%	1472	1.99%	/univ/nyushi/seikyul.htm
8	2072	0.04%	968	1.31%	/univ/nyushi/goukaku-suisen.htm
9	6296	0.13%	767	1.04%	/kagawa uni/sub1.htm
10	4664	0.10%	745	1.01%	/univ/nyushi/gakubu-m2.htm

図12 Top 10 of Exit Pages

Sites

サイトのトップ30はほとんど学内のサイトである。中で唯一の学外サイトが、“mail1.city.takamatsu.kagawa.jp.000a.232.7.211.in-addr.arpa”と変わった表示になっているが、これは、211.7.232.2のIPアドレスに対して、DNSが次のように答えるためである。おそらく、高松市役所内

の端末から一般の市民がアクセスしたものであろう。

```
2.232.7.211.in-addr.arpa. is an alias for
    2.000A.232.7.211.in-addr.arpa.
2.000A.232.7.211.in-addr.arpa. domain name pointer
    mail1.city.takamatsu.kagawa.jp.000A.232.7.211.in-addr.arpa.
```

Top 30 of 27795 Total Sites									
#	Hits	Files	KBytes	Visits	Hostname				
1	36912	0.77%	8346	0.29%	225044	0.71%	179	0.23%	proxy.kagawa-u.ac.jp
2	19033	0.40%	12038	0.42%	103565	0.33%	198	0.26%	133.92.4.134
3	17767	0.37%	10021	0.35%	85256	0.27%	205	0.27%	133.92.4.101
4	17439	0.37%	5668	0.20%	67928	0.21%	92	0.12%	mail1.city.takamatsu.kagawa.jp.000a.232.7.211.in-addr.arpa
5	17289	0.36%	170	0.01%	2236	0.01%	232	0.30%	133.92.81.108
6	15140	0.32%	8644	0.30%	74289	0.23%	157	0.20%	133.92.4.137
7	14978	0.31%	9622	0.33%	89250	0.28%	148	0.19%	133.92.4.132
8	14720	0.31%	10359	0.36%	102958	0.32%	136	0.18%	133.92.203.119
9	13579	0.28%	7357	0.25%	63629	0.20%	137	0.18%	133.92.5.126
10	13480	0.28%	8914	0.31%	81121	0.26%	151	0.20%	133.92.4.131
11	13274	0.28%	7925	0.27%	67769	0.21%	176	0.23%	133.92.5.101
12	13133	0.28%	7738	0.27%	66501	0.21%	172	0.22%	133.92.4.108
13	13107	0.27%	2996	0.10%	30057	0.09%	92	0.12%	133.92.57.202
14	13089	0.27%	7426	0.26%	62149	0.20%	141	0.18%	133.92.4.109
15	12845	0.27%	8530	0.29%	74144	0.23%	118	0.15%	133.92.203.126
16	12488	0.26%	7564	0.26%	62742	0.20%	146	0.19%	133.92.4.119
17	12481	0.26%	8345	0.29%	69120	0.22%	125	0.16%	133.92.203.104
18	12307	0.26%	7690	0.27%	65265	0.21%	140	0.18%	133.92.5.127
19	11740	0.25%	7854	0.27%	68400	0.22%	139	0.18%	133.92.4.135
20	11735	0.25%	7047	0.24%	67205	0.21%	125	0.16%	133.92.4.121
21	11718	0.25%	7613	0.26%	68133	0.21%	126	0.16%	133.92.203.130
22	11680	0.24%	7458	0.26%	61871	0.19%	148	0.19%	133.92.5.123
23	11643	0.24%	7828	0.27%	65216	0.21%	110	0.14%	133.92.203.101
24	11608	0.24%	6225	0.21%	57882	0.18%	131	0.17%	133.92.4.102
25	11547	0.24%	7332	0.25%	61134	0.19%	103	0.13%	133.92.203.112
26	11474	0.24%	6467	0.22%	60213	0.19%	118	0.15%	133.92.4.115
27	11404	0.24%	7490	0.26%	71029	0.22%	134	0.17%	133.92.4.124
28	11363	0.24%	6381	0.22%	52754	0.17%	137	0.18%	133.92.59.209
29	11360	0.24%	7370	0.25%	61085	0.19%	118	0.15%	133.92.203.116
30	11200	0.23%	7082	0.24%	60394	0.19%	128	0.17%	133.92.4.114

図13 Top 30 of Sites

Referrers

Referrerというのは、そのページを訪れた訪問者のリンク元ページのURLである(図15)。

Top 10 of 27795 Total Sites By KBytes									
#	Hits		Files		KBytes		Visits		Hostname
1	36912	0.77%	8346	0.29%	225044	0.71%	179	0.23%	proxy.kagawa-u.ac.jp
2	6934	0.15%	5463	0.19%	195850	0.62%	112	0.15%	nulgw11.unisys.co.jp
3	4476	0.09%	2846	0.10%	114090	0.36%	366	0.47%	msnbot.msn.com
4	19033	0.40%	12038	0.42%	103565	0.33%	198	0.26%	133.92.4.134
5	14720	0.31%	10359	0.36%	102958	0.32%	136	0.18%	133.92.203.119
6	890	0.02%	857	0.03%	100995	0.32%	5	0.01%	202.217.35.13
7	14978	0.31%	9622	0.33%	89250	0.28%	148	0.19%	133.92.4.132
8	17767	0.37%	10021	0.35%	85256	0.27%	205	0.27%	133.92.4.101
9	13480	0.28%	8914	0.31%	81121	0.26%	151	0.20%	133.92.4.131
10	15140	0.32%	8644	0.30%	74289	0.23%	157	0.20%	133.92.4.137

図14 Top 10 of Sites By KBytes

Top 30 of 2230 Total Referrers			
#	Hits		Referrer
1	3115886	65.25%	http://www.kagawa-u.ac.jp/
2	527283	11.04%	- (Direct Request)
3	165030	3.46%	http://www.kagawa-u.ac.jp/kagawa_uni/lnk.htm
4	160685	3.36%	http://www.kagawa-u.ac.jp/univ/nyushi/
5	99833	2.09%	http://www.kagawa-u.ac.jp/kagawa_uni/sub1.htm
6	70432	1.47%	http://www.kagawa-u.ac.jp/kagawa_uni/inf.htm
7	55703	1.17%	http://www.kagawa-u.ac.jp/kagawa_uni/sub4.htm
8	48421	1.01%	http://www.kagawa-u.ac.jp/kagawa_uni/acs.htm
9	46733	0.98%	http://www.kagawa-u.ac.jp/univ/nyushi/index.htm
10	36234	0.76%	http://www.kagawa-u.ac.jp/kagawa_uni/hot.htm
11	31491	0.66%	http://www.kagawa-u.ac.jp/univ/nyushi/seikyul.htm
12	24504	0.51%	http://www.kagawa-u.ac.jp/univ/nyushi/index.html
13	21768	0.46%	http://www.kagawa-u.ac.jp/kagawa_uni/msg.htm
14	21407	0.45%	http://www.kagawa-u.ac.jp/kagawa_uni/kou.htm
15	18893	0.40%	http://www.kagawa-u.ac.jp/index.html
16	16401	0.34%	http://www.kagawa-u.ac.jp/
17	15378	0.32%	http://www.kagawa-u.ac.jp/kagawa_uni/smp.htm
18	13471	0.28%	http://www.kagawa-u.ac.jp/univ/nyushi/shin.htm
19	13103	0.27%	http://www.kagawa-u.ac.jp/ccrd/header/index.html
20	12805	0.27%	http://www.kagawa-u.ac.jp/univ/nyushi/gakubu-m.htm
21	12388	0.26%	http://www.kagawa-u.ac.jp/kagawa_uni/act1.htm
22	11675	0.24%	http://www.kagawa-u.ac.jp/lifelong/
23	10932	0.23%	http://www.kagawa-u.ac.jp/kagawa_uni/sub2.htm
24	10416	0.22%	http://www.kagawa-u.ac.jp/kagawa_uni/sub3.htm
25	10378	0.22%	http://search.yahoo.co.jp/bin/search
26	10199	0.21%	http://www.kagawa-u.ac.jp/univ/nyushi/zisshil.htm
27	9184	0.19%	http://www.kagawa-u.ac.jp/isc/index-j.html
28	8959	0.19%	http://www.kagawa-u.ac.jp/kagawa_uni/sub5.htm
29	8101	0.17%	http://www.kagawa-u.ac.jp/univ/nyushi/gakubu.htm
30	8080	0.17%	http://www.kagawa-u.ac.jp/kagawa_uni/act2.htm

図15 Top 30 of Referrers

Search

検索文字の統計である。ほとんどの検索サイトでは、検索キーワードを URL の一部として受け取っている。したがって、訪問者がその結果を見てから香川大学を訪れた場合は Referer フィールドにその検索文字が残っているのである。先に説明したように、ブラウザがエンコードした文字をできるだけ復元するように工夫はしたのだが、19位の“紊 y□”というわけのわからない文字は、実は「大学」である。もとの文字は UTF-8でエンコードされていたのだが、何故かコード変換に使用した Jcode.pm というライブラリーが間違っていて変換している。Referrer の統計から推定すると、利用された検索サイトの割合は、1位が Yahoo で65%、2位が Google で23%、3位が msn で8%、その他が4%であった。

表の最後に、View All Search Strings というリンクがある。これはすべての検索文字の統計ページへのリンクである。

Top 20 of 1475 Total Search Strings			
#	Hits		Search String
1	12414	76.91%	香川大学
2	738	4.57%	香川大
3	103	0.64%	香川大学
4	96	0.59%	香川
5	86	0.53%	yahoo
6	61	0.38%	kyoutei
7	57	0.35%	kagawa
8	37	0.23%	香川大学 合格発表
9	28	0.17%	文教ニュース
10	26	0.16%	kagawa university
11	24	0.15%	東洋炭素
12	21	0.13%	地域新生コンソーシアム
13	21	0.13%	香川大学大学院
14	21	0.13%	香川大教育学部
15	19	0.12%	http://www.kagawa-u.ac.jp/
16	18	0.11%	紺綬褒章
17	16	0.10%	u
18	16	0.10%	かがわだいがく
19	15	0.09%	紊 y ◆
20	15	0.09%	香川大学 岩月

[View All Search Strings](#)

図16 Top 20 of Search Strings

Agents

Agents は接続してきたクライアントプログラムを表す。一般にはウェブブラウザを表すことが多いが、検索ロボットであることも少なくない。実は、筆者の運用しているロボットが毎日1回ページをアクセスしているはずであるが、これは除いてある。その他のロボットはそのまま集計している。

Top 15 of 325 Total User Agents		
#	Hits	User Agent
1	4071941	85.27% MSIE 6.0
2	244434	5.12% Mozilla/5.0
3	172189	3.61% MSIE 5.5
4	123077	2.58% MSIE 5.0
5	40439	0.85% Mozilla/3.01 (compatible;)
6	20407	0.43% MSIE 5.2
7	17647	0.37% MSIE 5.1
8	16331	0.34% Mozilla/4.7
9	13667	0.29% MSIE 4.0
10	6887	0.14% Googlebot/2.1 (+http://www.google.com/bot.html)
11	5634	0.12% msnbot/0.3 (+http://search.msn.com/msnbot.htm)
12	4429	0.09% Yahoo! Slurp
13	3227	0.07% HyperRobot
14	3060	0.06% Mozilla/4.0
15	2271	0.05% Mozilla/4.0 (compatible;)

図17 Top 15 of User Agents

Countries

クライアントのホスト名からその国名を推定したものである。Unresolved/Unknown が多いのは、学内からアクセスしてくる多くの IP アドレスが DNS に登録されていないため、その所属する国がわからないためであろうと推測される。ただし、国によっては、ドメイン名を他国の人に販売していることがあるので、注意が必要である。

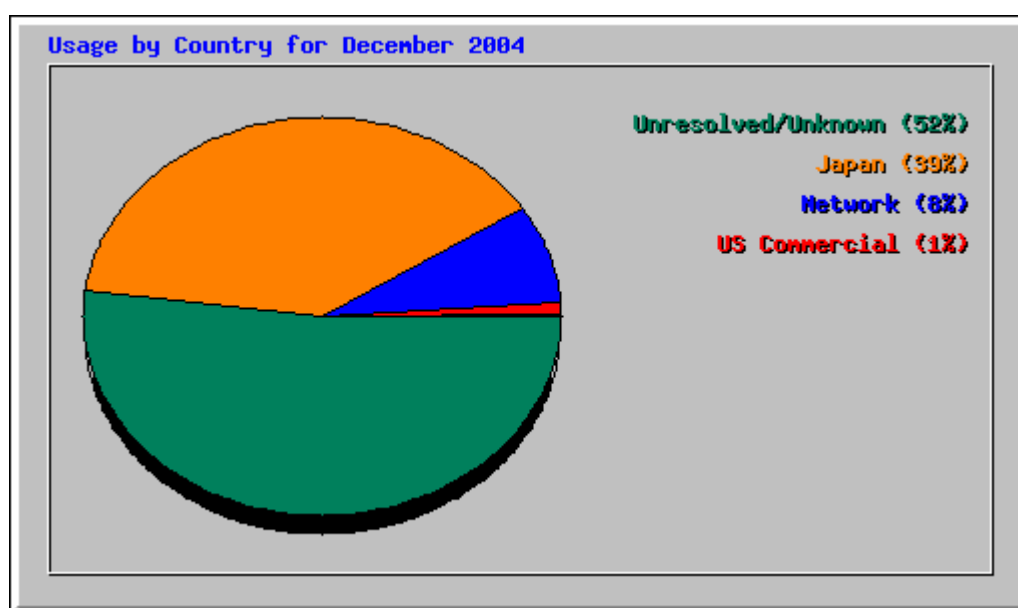


図18 Usage by Country for December 2004

Top 30 of 63 Total Countries							
#	Hits		Files		KBytes		Country
1	2468713	51.70%	1405145	48.47%	13494953	42.44%	Unresolved/Unknown
2	1844646	38.63%	1216359	41.96%	14399033	45.28%	Japan
3	376080	7.88%	254750	8.79%	2973236	9.35%	Network
4	55943	1.17%	33345	1.15%	607025	1.91%	US Commercial
5	17519	0.37%	5746	0.20%	68633	0.22%	Old style Arpanet (arpa)
6	2132	0.04%	1860	0.06%	76666	0.24%	Germany
7	1949	0.04%	1625	0.06%	65922	0.21%	US Educational
8	1665	0.03%	631	0.02%	5566	0.02%	Tuvalu
9	789	0.02%	737	0.03%	7875	0.02%	Taiwan
10	753	0.02%	684	0.02%	6336	0.02%	United Kingdom
11	644	0.01%	482	0.02%	5980	0.02%	Non-Profit Organization
12	473	0.01%	325	0.01%	3059	0.01%	China
13	452	0.01%	340	0.01%	3461	0.01%	New Zealand (Aotearoa)
14	438	0.01%	331	0.01%	4029	0.01%	Netherlands
15	345	0.01%	338	0.01%	7213	0.02%	Canada
16	329	0.01%	240	0.01%	4085	0.01%	Australia
17	295	0.01%	237	0.01%	15952	0.05%	France
18	231	0.00%	224	0.01%	1582	0.00%	Mexico
19	213	0.00%	206	0.01%	1611	0.01%	Belgium
20	171	0.00%	171	0.01%	5115	0.02%	Hong Kong
21	166	0.00%	157	0.01%	1836	0.01%	Poland
22	163	0.00%	163	0.01%	1298	0.00%	Denmark
23	147	0.00%	146	0.01%	1350	0.00%	US Government
24	142	0.00%	139	0.00%	1946	0.01%	Thailand
25	101	0.00%	98	0.00%	2495	0.01%	Sweden
26	97	0.00%	97	0.00%	2267	0.01%	US Military
27	90	0.00%	53	0.00%	1100	0.00%	Singapore
28	74	0.00%	70	0.00%	975	0.00%	Nepal
29	70	0.00%	68	0.00%	529	0.00%	Finland
30	65	0.00%	65	0.00%	476	0.00%	Pakistan

図19 Top 30 of Countries

3. Analog

Analog は、Stephen Turner が開発しているオープンソースのウェブアクセスログ解析ツールである。この人は現在、Analog の商用版として、より機能を強化した「ClickTracks」という製品を作り、その会社の中心的人物となっている。実は Analog の方が Webalizer より歴史があり、Apache の環境では定番と言われている。フリーの Analog は、現在も開発が続けられており、「日本 Analog ユーザー会」により日本語への対応やマニュアルの翻訳などが行われている。今回は出力を一見して、Webalizer の方が見栄えが良かったので、こちらの方の調査は十分ではないのだが、簡単に紹介しておく。

まず、Debian では、“`apt get install analog`” とするだけでインストールできた。そこで、設定ファイルの `/etc/analog.cfg` でログファイルだけを変更して、適当なディレクトリーで Analog を実行すると、その中に `Report.html` 他、いくつかの png 形式のイメージファイルが作成された。以下にそのイメージを示す。結果の中に棒グラフが含まれているが、実際の色は赤である。



Web Server Statistics for Debian Linux System

Program started at Sun-30-Jan-2005 20:45.

Analysed requests from Wed-29-Sep-2004 21:01 to Sun-30-Jan-2005 20:45 (122.99 days).

General Summary

(Go To: [Top](#): General Summary: [Monthly Report](#): [Daily Summary](#): [Hourly Summary](#): [Domain Report](#): [Organisation Report](#): [Search Word Report](#): [Operating System Report](#): [Status Code Report](#): [File Size Report](#): [File Type Report](#): [Directory Report](#): [Request Report](#))

This report contains overall statistics.

(Figures in parentheses refer to the 7-day period ending 30-Jan-2005 20:45).

Successful requests: 762,986 (56,588)

Average successful requests per day: 6,203 (8,083)

Successful requests for pages: 99,675 (8,161)

Average successful requests for pages per day: 810 (1,165)

Failed requests: 11,052 (397)

Redirected requests: 827 (47)

Distinct files requested: 78,527 (8,713)

Distinct hosts served: 6,652 (646)

Corrupt logfile lines: 2

Data transferred: 45.54 gigabytes (696.28 megabytes)

Average data transferred per day: 379.17 megabytes (99.47 megabytes)

図20 Analog の General Summary 英語版

ウェブサーバの統計 www.kagawa-u.ac.jp

統計開始日時: 2005年 1月26日(水) 22時56分.

解析対象期間: 2005年 1月23日(日) 04時03分 ~ 2005年 1月26日(水) 22時56分 (3.79 日).

全体の概要

(Go To: [先頭](#): [全体の概要](#): [月別レポート](#): [曜日別集計](#): [時間別集計](#): [ドメインレポート](#): [組織別レポート](#): [検索語レポート](#): [OSLレポート](#): [状態コードレポート](#): [ファイルサイズレポート](#): [ファイル種類別レポート](#): [ディレクトリレポート](#): [リクエストレポート](#))

このレポートは、全体の統計を表します。

リクエスト成功件数: 1,287,822
上記の日別平均: 340,081
ホームページへのリクエスト成功件数: 59,451
上記の日別平均: 15,699
リクエスト不成功件数: 9,584
リダイレクトされたリクエスト件数: 153
リクエストされたファイル数: 2,107
サービスホスト数: 7,933
異常ログ行数: 4
データ転送量: 7.452 G バイト
上記の日別平均: 1.967 G バイト

図23 Analog の General Summary 日本語版

月別レポート

(Go To: [先頭](#): [全体の概要](#): [月別レポート](#): [曜日別集計](#): [時間別集計](#): [ドメインレポート](#): [組織別レポート](#): [検索語レポート](#): [OSLレポート](#): [状態コードレポート](#): [ファイルサイズレポート](#): [ファイル種類別レポート](#): [ディレクトリレポート](#): [リクエストレポート](#))

このレポートは、月別の統計を表します。

1単位 (+) は 2,500 ページのリクエスト あるいはその値未満を表す。

月:	リクエスト数:	ページ数:	
-----:	-----:	-----:	
2005年 1月:	1287822:	59451:	+++++

最も混雑した月: 2005年 1月 (59,451 ページのリクエスト).

図24 Analog の Monthly Summary 日本語版

曜日別集計

(Go To: [先頭](#): [全体の概要](#): [月別レポート](#): [曜日別集計](#): [時間別集計](#): [ドメインレポート](#): [組織別レポート](#): [検索語レポート](#): [OSレポート](#): [状態コードレポート](#): [ファイルサイズレポート](#): [ファイル種別レポート](#): [ディレクトリレポート](#): [リクエストレポート](#))

このレポートは、全週の曜日別総和の統計を表します。

1単位(+)は500ページのリクエストあるいはその値未満を表す。

```

日: リクエスト数: ページ数:
--: -----: -----:
日:      206673:      10347: ++++++
月:      365296:      16868: ++++++
火:      369095:      17110: ++++++
水:      346758:      15126: ++++++
木:         0:         0:
金:         0:         0:
土:         0:         0:
    
```

図25 Analog の Weekly Summary 日本語版

また、検索語の解析は次のように表示された。出力はEUCになっているので、unicode文字が化けているようである。

上位 30 検索単語の表: 基準は リクエスト数, ソート順は リクエスト数 .

```

リクエスト数: 検索単語
-----: -----
3273: 香川大学
237: 香川大
58: u.ac.jp/
57: cache:qsJ2
57: ZD7E58J:www.kagawa
55: 香川
50: 紊 y ⑥□□綺□□宴□□□∞□□
42: kagawa
40: 大学
40: FxT2RkJ:www.kagawa
40: u.ac.jp/kagawa_uni/his.htm
40: cache:mAcP
36: スウェーデン
35: 薙□綏□紊 y ⑥□□罐Ⓣ□Ⓣ□□紊 y ⑥
23: university
18: 国立大学
14: 健康と運動
13: ジェンダー
13: kyoutei
13: cache:ojJU35W45rwJ:www.kagawa
13: edu/center3.html
13: u.ac.jp/high
    
```

- 12: 授業料
- 10: かがわだいがく
- 9: 薙□綏□紊 y ⑥
- 9: 入試情報
- 9: 入試
- 9: 罩□□□□□□□
- 9: /x8d/x81/x90/xec/x91/xe5/x8aw
- 9: 募集要項
- 1265: [その他: 885 検索語句]

4. むすび

学内のウェブサイトは、現在、大学トップページからリンクされているものだけで、72が確認されている。今後ますますウェブによる情報発信が増加していくものと思われるが、はたしてその利用実態が十分に把握されているのであろうか。本稿で紹介したようなツールを利用することにより、アクセスの概要が簡単に把握できるのであるから、ウェブ管理者には何らかの解析ツールを導入することを勧めたい。今後の課題としては、利用者の動きを追跡してみたい。推測であるが、大学のウェブページは必ずしも使いやすくなっていないようである。おそらく、必要な情報を求めて、あちこちさまよっている利用者がいるのではないかと考えている。そういう実態が明らかになれば、何らかの改善策を提案できるであろう。

参考文献

1. いなかどっとコム・石井 研二 著、『アクセスログ解析の教科書』、翔泳社、2004年9月
2. 松木 孝幸・太田 俊哉・高橋 基信 (著)、『Analog—Web アクセスログの高速解析ソフト』、九天社、2003年3月
3. Webalizer のウェブサイト <http://www.mrunix.net/webalizer/>
4. Webalizer による Web アクセスログ解析 <http://acorn.zive.net/~oyaji/www/webalizer.htm>
5. webalizer の導入(apache のログ解析) <http://penguin.nakayosi.jp/linux/webalz.html>
6. Analog のウェブサイト <http://www.analog.cx/>
7. 日本 Analog ユーザー会 <http://www.tokyo-kasei.ac.jp/~matsuki/analog/>
8. アクセスログを解析する「Analog」 <http://baseband.ne.jp/free/bff212-20040202.html>

香川県における医療 IT への取り組みと今後の展望

医学部附属病院

原 量宏

先日2月9日の日経新聞一面トップに、電子カルテ導入の義務化が大きく載っているように、医療のIT化、特に電子カルテや遠隔診断のネットワークに関してマスコミでもしばしば取り上げられる様になっています。政府は平成13年年1月に内閣直属のIT戦略本部において「e-Japan戦略」を策定し、平成17年までに日本が世界最先端のIT国家となることを大きな目標に掲げています。

厚生労働省は「保健医療分野の情報化にむけてのグランドデザイン」をまとめ、その中でこれから改善すべき「医療の課題」として、診療情報の提供、医療の質の向上、効率化、安全確保を重点項目と設定し、その課題を解決するための方策として、電子カルテシステム、レセプトコンピュータ、EBM支援、遠隔医療支援、オーダリングシステム、個人・資格認証システム、物流管理システムを掲げています。これまでの病院情報システムは、大規模な病院における導入が中心でしたが、最近は中小の病院においてもオーダリングシステムが導入され、小規模な診療所においても診療報酬請求を目的としたシステム(レセコン)が急速に普及しています。従来から電子カルテを論ずる場合、個々の電子カルテの機能向上が中心課題でしたが、他の電子カルテとの相互連携に関しては、あまり考慮されない傾向にありました。電子カルテが情報ネットワークで相互に結ばれ、地域全体を一体として利用することができれば、電子カルテの機能は飛躍的に高まることとなります。

さいわい香川県においては、県、医師会、大学の協力のもとに、産婦人科の医療機関をむすぶ周産期ネットワーク、さぬき市(旧長尾町、寒川町、大川町)と附属病院の間をネットワークでむすぶ独居高齢者の健康管理システムが稼働しています。なかでも特記すべき事は、平成14年6月に画期的な画像センター構想、「かがわ遠隔医療ネットワーク(Kagawa Medical Internet Exchange, K-MIX)」がスタートしたことです。本ネットワークは香川県の一般財源で実現したもので、全県的な取り組みとしては全国でもはじめてのもので、県と県医師会、そして香川大学医学部が一体となって運営していることが大きな特徴です。かがわ遠隔医療ネットワークでは医療機関がインターネット環境下であれば、容易に医用画像を相互に伝送できる仕組みになっています。すでにDICOM対のCT、MR等の機器をもっている施設は、これまで通りDICOM規格で送受信できることが特徴で、またDICOM対応の機器をもたない医療機関においても、JPEG規格などの画像(顕微鏡による病理画像、内視鏡画像、眼底、心電図、デジカメによる皮膚の画像等)をセンターに送れば、DICOM規格に変換されたのちにセンターのDICOMサーバ保存されます。画像情報以外にも診療情報も送受信することが可能で、将来の電子カルテとの連携も視野にいられています。セキュリティに関しては、SSLにくわえ、UMINの医療用VPNを導入済みで、将来的には保健医療分野の電子認証局(HPKI)の導入も考慮しています。実際の利用に際しては、まず診断を依頼する医師が診断依頼情報をK-MIXの画面に入力して、依頼先の医師あてに画像とともに送信します。依頼された医師に電子メールが届くので、その画面のなかのURLをクリックすると、診断依頼内容と画像が表示されます。診断結果を入力しさらにクリックすれば、依頼した元の医師へメールが送信されます。診断を依頼した医師はメールを開きURLを指定すると診断結果がキー画像とともに表示されるようになっており非常に使いやすいシステムとなっています。かがわ遠隔医療ネットワークが核となり、将来全国すべての医療機関がネットワーク化される時代がくることを期待しています。

医学部池戸地区基盤センター無線 LAN 接続システム・研究支援システム

医学部 木村 正司

基盤センター医学部地区では、旧香川医科大学情報メディアセンターで展開された事業を継承した。これら事業(平成16年年報参照)は地区ネットワーク基盤整備の他、大学で求められる実際の教育・実習・研究及び医療の場でネットワーク基盤を介した効率的な運営が図られるように、事業責任者のもと計画実施されている。

平成15年度にはセンター別予算から、講義実習棟および臨床講義棟の講義室全室、学生会館などにアクセスポイントを設置し、基盤センター管轄の学内ネットワークに無線 LAN 接続 (IEEE802.11a,b,g 対応) できる環境を構築し、設置型 PC の不足に対応している。

学術データ発表支援システムでは、大型ポスター用プリンタ (B0 プリンタ) も備えており学部外からの利用も可能である。何れも医学部臨床研究棟 5 F に設置されている。



A3フラットベツトスキャナ



ドキュメントスキャナ



X-RAYフィルムスキャナ



フィルムスキャナ



ノンリニアビデオ編集



デジタル文書作成システム



画像解析システム



フィルムレコーダ



写真タイププリンター



B0プリンタ



A3レーザープリンタ(モノクロ・カラー)



認証 DHCP サービスの利用状況について

工学部(信頼性情報システム工学科)

最所圭三

総合情報基盤センター

瀬野芳孝

曽根計俊

香川大学においては、平成 15 年 3 月に教育用計算機システムの更新に伴い認証 DHCP サーバを導入した。本報告では、認証 DHCP サービスの仕組みを説明したあと利用状況について報告する。

1. 認証 DHCP サービスについて

認証 DHCP サービスを説明する前に、DHCP サービスの仕組みについて説明する。

DHCP とは、Dynamic Host Configuration Protocol の頭文字で、動的にコンピュータのネットワークインタフェースの設定を行いものである。ここでコンピュータではなくネットワークインタフェースと言っているのは、1 つのコンピュータに対して複数のネットワークインタフェースを持つことができるためである。事実、最近のノート PC では、有線 LAN のネットワークインタフェースと無線 LAN のネットワークインタフェースの両方を持つものがある。DHCP においては、コンピュータをネットワークに接続するのに必要な設定を行う。多くの場合、インターネット中でコンピュータを識別するための 32 ビットの番号である IP アドレス、直接通信が可能な範囲 (セグメント) を示すサブネットマスク、サーバ等の名前から IP アドレスに変換を行う DNS サーバの IP アドレス、直接通信が可能でない時の中継してくれるルータの IP アドレス (通常は 1 個でデフォルトゲートウェイと呼ばれる) を設定する。なお、IPv6 と呼ばれる次世代の IP アドレスでは 128 ビットの番号を用いる。

図 1 は、DHCP による IP アドレスの割当ての手順を示す。ここでは、IP アドレスの割当てのみ示しているが、 の IP アドレスの使用許可を送る際に他の情報も送られる。これ以降、PC 等を「ホスト」と呼ぶ。

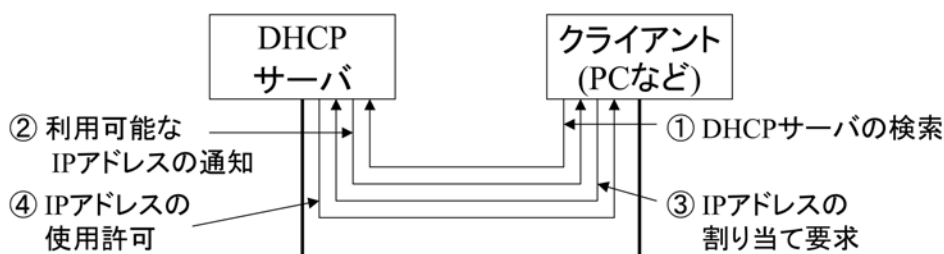


図 1 DHCP による IP アドレスの割当て手順

IP アドレスの割当てを要求するホストは、最初に図 1 の で示す「DHCP サーバの検索」を要求する。この要求は、同一セグメント内の全ホストが受け取るブロードキャストを用いて、パケットとしてネットワークに送られる。このとき、そのパケットにはネットワークインタフェースを識別するために、全

てのネットワークインタフェースが持っている固有の番号である **MAC アドレス** と呼ぶ 48 ビットの番号をパケットに含める。同一セグメント内のホスト間では、最終的にはこの MAC アドレスを用いて互いに通信する。同一セグメント内に DHCP サーバが存在すれば、そのパケットを受け取り、割当て可能な IP アドレスがあればそれを通知する ()。もし、複数の DHCP サーバが同一セグメント内に存在すれば、それぞれ割当て可能な IP アドレスを、要求を出してきたホストに返す。このパケットを受け取ったホストは、通知された IP アドレスを指定して、IP アドレスの割当て要求を行う ()。この要求を受け取った DHCP サーバは、その要求が問題なければ IP アドレスの使用許可を示すパケットを要求したホストに送る ()。このパケットを受け取ったホストは、自分のネットワークインタフェースを受け取った情報に従って設定する。

割り当てられた IP アドレスは永久に使用できるものでなく、期限が付いている。この期限のことをリースタイム (lease time) と呼ぶ。香川大学では、リースタイムはセグメントにより 30 分または 10 分に設定されている。DHCP により IP アドレスが割り当てられているホストはリースタイムが切れる前にリースタイムの更新を行う必要がある。この更新は、リースタイムの半分になったときに行われる。

1 つのセグメントに 1 つの DHCP サーバを置くことは、DHCP サービスを行うセグメントの数だけ DHCP サーバが必要になり非効率であると共に、多くの DHCP サーバを管理しなければならないので管理のオーバーヘッドも大きい。このため、各セグメントに DHCP relay agent と呼ばれるホストを置くことにより、この問題を解決している (図 2)。DHCP relay agent はブロードキャストされた DHCP 関係のパケットを異なるセグメントに置いてある DHCP サーバに転送し、その応答を要求を出したホストに返す。通常は、図 3 に示すように、ルータにこの機能を持たせることにより、特殊なホストを置かなくて済むようにしている。

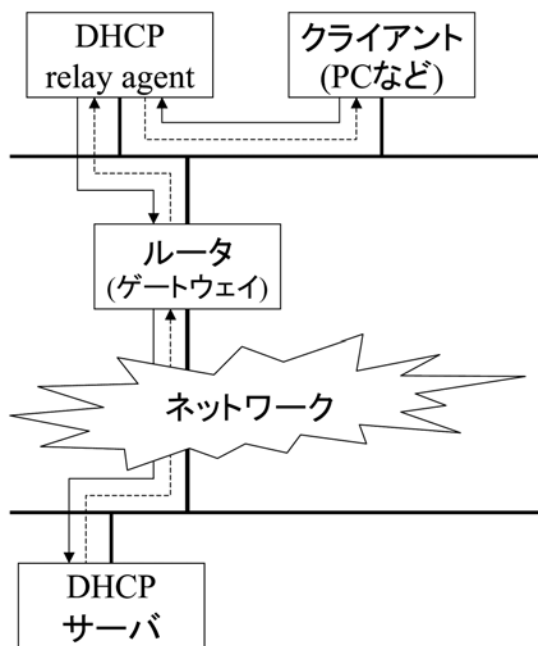


図 2 DHCP relay agent による DHCP サーバの共有

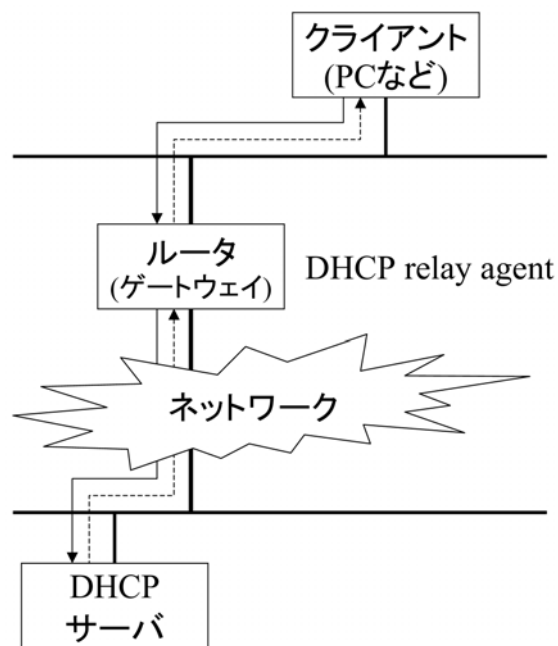


図 3 ルータを DHCP relay agent として用いる

次に、認証 DHCP サービスについて説明する。香川大学では、DHCP において IP アドレスを発行する際にユーザ認証を行うために、QIP DHCP サーバを導入した。QIP DHCP サービスでは、図 4 に示すように、IP アドレスを与えて良いネットワークインタフェースの MAC アドレスとその所有者をデータベース化した認証サーバと、クライアントからの DHCP サーバの検索が入ったときに、認証サーバと呼ぶサーバにそのインタフェースに IP アドレスを与えて良いかどうかを問い合わせる機能を

追加した認証 DHCP サーバを用いる。認証 DHCP サーバからの問い合わせに対して認証サーバはデータベース内に問い合わせのインタフェースの MAC アドレスが登録されていれば許可を与える。認証 DHCP サーバは許可が返ってくれば、通常の IP アドレスを与える。不許可であれば、ネットワークインタフェースの MAC アドレスの登録するための Web サーバにアクセスできる認証用の IP アドレスを与える。この認証用の IP アドレスで通信できる範囲を制限することで、ネットワークセキュリティを向上させている。

このシステムを利用するためには、認証サーバにユーザ登録をしなければならない。現在、認証サーバには総合情報基盤センターに登録されている全ての利用者が登録されているので、学生は自動的に、教官は総合情報基盤センターに利用者登録することで利用できる。MAC アドレスが登録されていないネットワークインタフェースを持つホストをネットワークに接続すると、認証用の IP アドレスが割り当てられるので、このアドレスを用いて認証用の Web サーバにアクセスする。認証用の Web サーバは、アクセスしてきた IP アドレスから認証 DHCP サーバの割当て情報により自動的に MAC アドレスを認識できるので、認証用の画面ではユーザ名とパスワードを入力するだけでよい。この認証が成功すれば、次回から外部と通信できる IP アドレスが割り振られる。

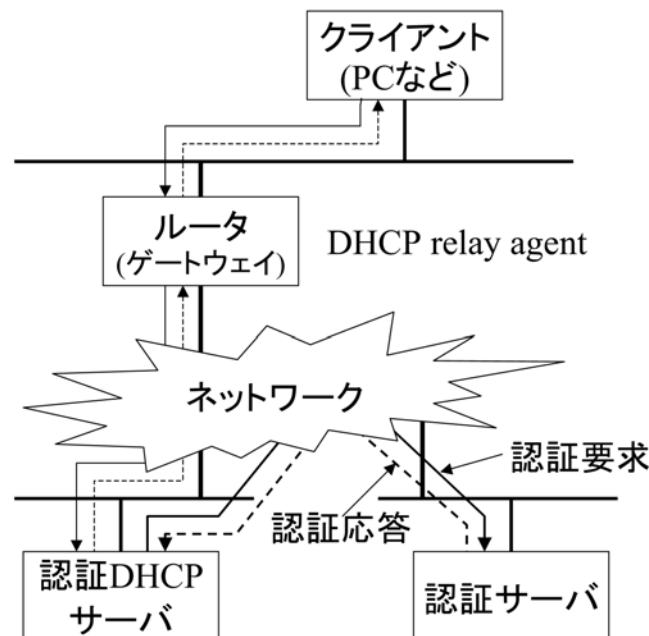


図4 認証 DHCP サービスの仕組み

2. 認証 DHCP サーバの利用状況

認証 DHCP サービスを実現するためにはルータの設定変更が必要であった。香川大学に入っているルータの仕様を調査等に数ヶ月かかり、2003年7月25日に実験を行い、成功を収めた。このあと、DHCP サービスを先行させていた工学部で、一部のセグメントで2003年9月末から試験運用を開始し、11月初旬に本格運用を開始した。その後、順次幸町、農学部の順に運用が開始された。工学部を先行させた理由は、工学部では既に DHCP サービスを行っておりセキュリティの面で早急に認証 DHCP サービスに移行する必要があったためである。計画当時は統合前だったため、医学部は入っていない。

以下は、昨年1月1日から12月31日までの IP アドレスの割当てログを元に認証 DHCP サービスの利用状況の総計処理を行った結果を示す。

筆者らは、認証 DHCP サーバから出力される膨大な割当てログから必要な情報を抽出するプログラムを作成し、それを用いて割当てログを生成した。工学部の認証 DHCP サーバには UNIX マシ

ンが、他の学部では Windows マシンが使われていた関係上、別々に開発した。UNIX マシンに対しては最所が、Windows マシンに対しては曾根が開発を行った。得られる情報は、いつどの MAC アドレスに対してどの IP アドレスが割り当てられたかである。

2.1 1日での最大/最小割当数の変遷

図5, 6, 7 は1日でのアドレスの割当ての最大数と最小数をグラフにしたものである。幸町地区の割当てログは、2004年1月7日から始まっているが、これは運用を示す日時でなく、IPアドレスの割当てログを収集できるようになり、その時採取できたログの開始時刻である。農学部では、2004年2月16日から始まっているが、これは農学部への認証 DHCP サーバの設置が遅れたためである。

まず幸町地区と農学部地区について見てみる。図5に幸町、図7に農学部の結果を示す。

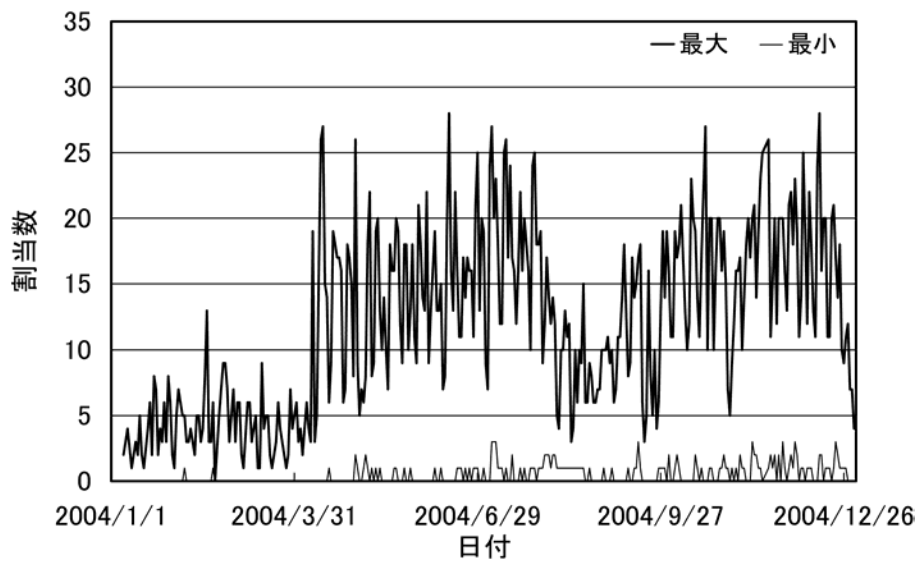


図5 IPアドレスの割当ての最大数と最小数の変遷 (幸町)

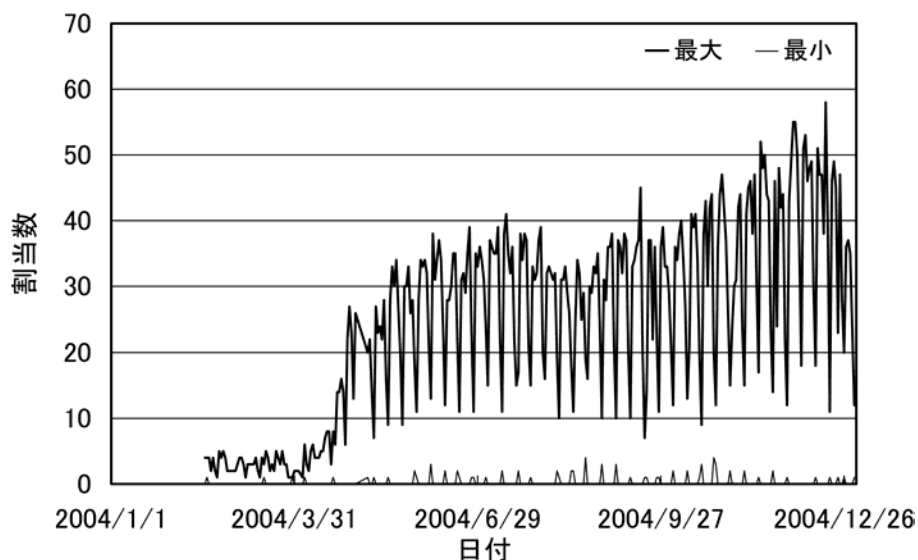


図6 IPアドレスの割当ての最大数と最小数の変遷 (農学部)

幸町や農学部では初めて導入されたこともあり、2004年3月まではほとんど利用されていないが、4月にならってから利用数が増えている。農学部には、はっきりとした増加傾向が見られる。次に工学部の結果を図7に示す。

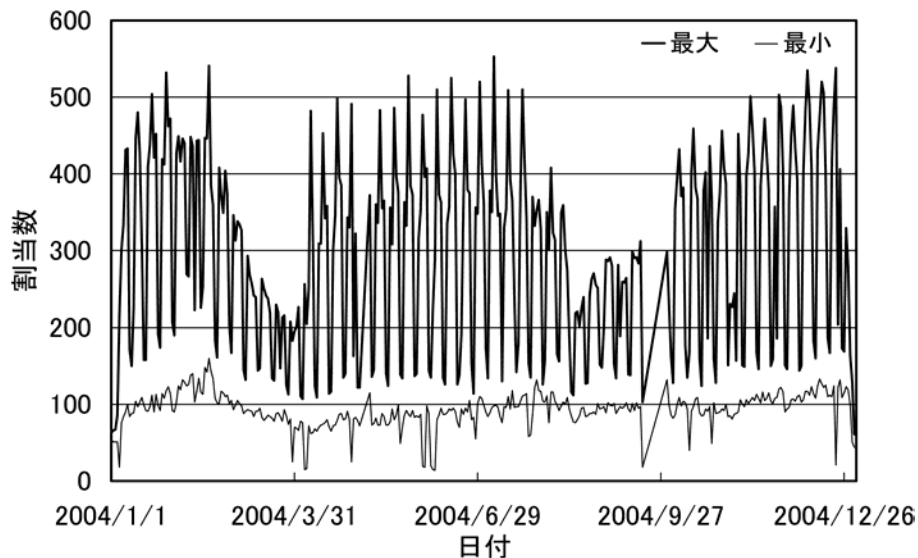


図7 IPアドレスの割当ての最大数と最小数の変遷 (工学部)

工学部においては、認証 DHCP サーバの導入前から DHCP サービスを行っており情報コンセントなどの環境が整っていることと学生全員がノート PC を持っていることにより、他の地区と比べ 10 倍以上の割当数になっている。また、年末年始、年度末、夏期休暇を除いてまんべんなく利用されているのが分かる。

夏期休暇の利用状況を見ると、幸町や工学部では落ち込みが大きいですが、農学部ではそれほどでもない。詳しく調べてみなければ分からないが、研究分野の違いや利用層の違いが出ているものと思われる。工学部では低学年から利用しているので、夏期休暇中の利用の落ち込みが大きくなる。幸町地区は母数が少ないので何とも言えないが、落ち込んだ分は工学部の学生が幸町で利用していた分だったかもしれない。

工学部以外は最小割当数が 0 にまで落ちるが、工学部では、一部の期間を除き最低でも 70 以上になっている。最低数が 70 以上ある理由は、学生が夜遅く閣内にいるのではなく、研究室の常時稼働しているホストに対しても DHCP により IP アドレスを割り当てているためである。また、最小数が著しく少なくなる理由は、認証 DHCP サーバあるいは DHCP relay agent であるルータの調子が悪くなることである。このため、工学部では、この最低数がある値（現在のところ 50）を下回る場合に DHCP のサービスを行っているプログラムを自動的に再起動するようにしている。

工学部と農学部では、一週間のリズムが規則正しくなっていることが分かる。逆に幸町では、一週間のリズムが全くないわけではないが、工学部や農学部ほど顕著に表れていない。情報コンセントの設置場所に依存しているのかもしれない。あるいは、町中にあり下宿から近いなどの理由もあるかもしれない。

2.2 曜日あたりの割当数の変遷

利用状況が安定していると思われる期間における曜日あたりの割当数を調べるために、ある週の割当数の変遷を調べてみた。図 8, 9, 10 は、2004 年 11 月 28 日 (日) から 2004 年 12 月 5 日 (日) までの割当数の変遷を示す。

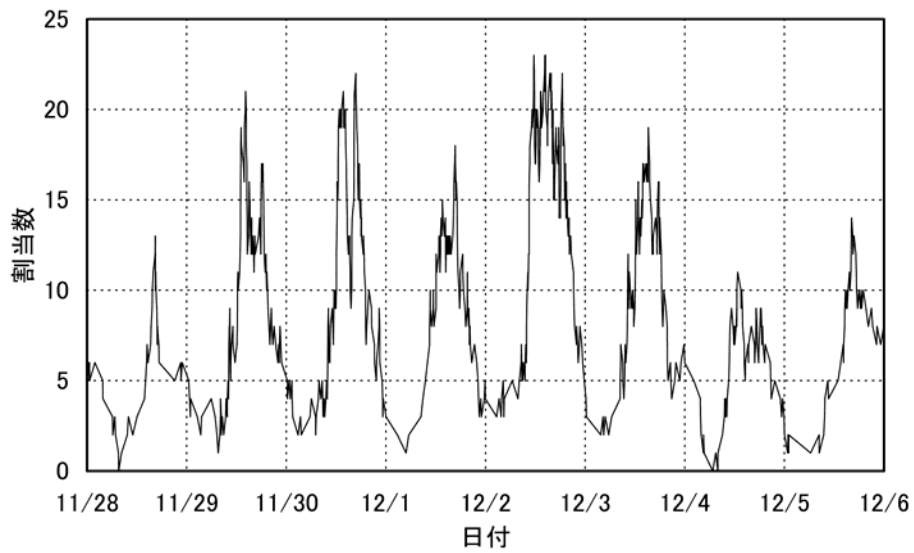


図8 IPアドレスの割当数の変遷(幸町)

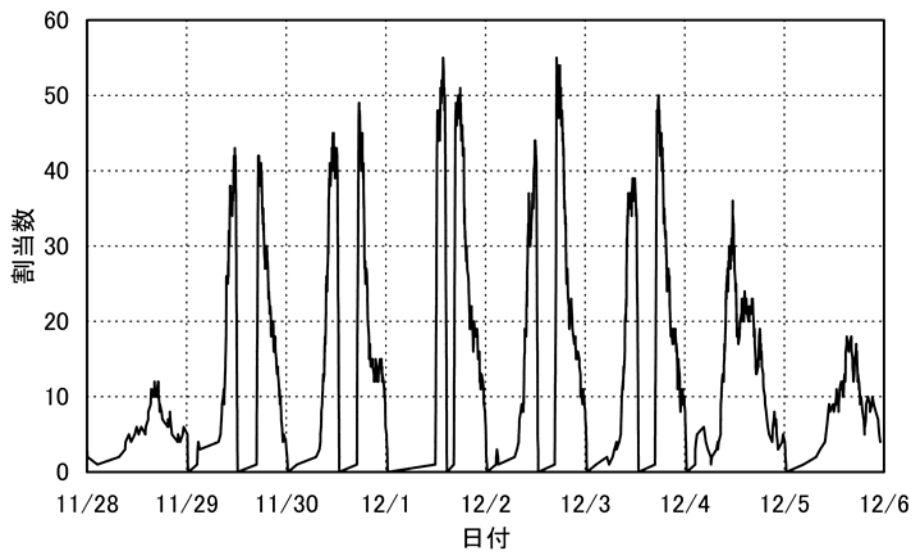


図9 IPアドレスの割当数の変遷(農学部)

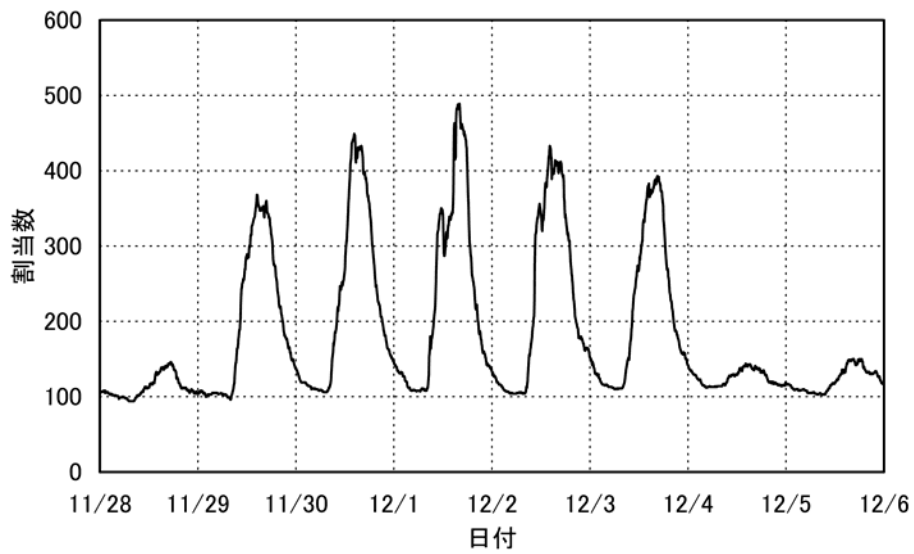


図10 IPアドレスの割当数の変遷(工学部)

どの地区も昼過ぎあたりにピークがあることが分かる。特徴的なのは、農学部である。他の地区が連続的に変化しているのに対して、農学部では、平日は3、4校時に当たる時間の利用数が0にまで落ちている。これは、DHCP サービスを行っている情報コンセントが授業で利用される部屋にあるためだと思われる。その証拠として、土日は連続して利用数が変化している。

各曜日の変化を見ると、工学部では水曜日の割当数が他の曜日に比べ多くなっている。これは、水曜日に1、2年生の授業が工学部で行われるためである。逆に幸町では土日を除けば、水曜日が最も割当数が少なくなっていることが分かる。

3. まとめ

以上、認証 DHCP サービスの利用状況を地区毎に調査した。まだまだ、調査期間が短く、1年を通じた傾向を見るには、あと1～2年は必要である。また、今回は、地区毎の統計処理を行ったが、今後はセグメント毎の統計情報も調べてみたいと思う。

今後の利用状況については、工学部では、学生全員がノート PC を携帯しているので、今後、新たに学生の利用者数が増加することはないが、研究室に少しずつパソコンが導入されているのでその分の増加が見込まれる。他地区については、情報コンセントの増設等のイベントが発生すれば利用数の増加が見込まれる。

業務報告

平成15年度利用申請一覧(研究・事務)

	所 属	氏 名	研 究 課 題	備 考
1	教育学部	青 木 昌 三	物理学に於けるいくつかの問題のシミュレーション	
2	教育学部	永 尾 智	英語助動詞の歴史的研究	
3	教育学部	土 屋 盛 茂	論理実証主義およびポパー哲学の研究	
4	教育学部	安 西 一 夫	情報交換とフーリエ解析の応用	
5	教育学部	黒 田 勉	研究	
6	教育学部	田 北 晋 一	図形処理	
7	教育学部	佐 藤 明 宏	国語科教育に関する研究	
8	教育学部	高 井 忠 昌	立方格子における分散曲線	
9	教育学部	日 野 陽 子	PPPダイヤルアップ	
10	教育学部	森 征 洋	風の局地性に関する研究	
11	教育学部	森 征 洋	宇宙・地球科学データベース	
12	教育学部	加 藤 みゆき	茶の風味に関する研究	
13	教育学部	山 本 茂 喜	国語科単元学習における評価法の研究	
14	教育学部	宮 崎 英 一	計算機による光学パターンの解析	
15	教育学部	小 池 和 男	素粒子の世代構造の研究	
16	教育学部	西 岡 圭 子	教育方法に関する研究	
17	教育学部	内 藤 浩 忠	整数論	
18	教育学部	藤 元 恭 子	幼児の身体表現に関する研究	
19	教育学部	Paul Batten	Ethnography of Compensatory Communicative Strategies	
20	教育学部	秋 山 智	グラフィックデザインにおけるコンピュータ処理	
21	教育学部	松 本 康	教科教育に関する調査研究	
22	教育学部	松 島 欣 哉	New England Transcendentalists, 特 に Henry David Thoreauの研究	
23	教育学部	青 山 夕 夏	情報交換	
24	教育学部	深 石 博 夫	数学の研究及び通信	
25	教育学部	松 村 雅 文	光散乱の計算と、その星間現象への応用	
26	教育学部	山 下 直 子	日本語・日本事情教材の作成の資料収集・連絡のため	
27	教育学部	小 川 育 子	親水性ポリマーの膨潤	
28	教育学部	村 山 聡	経済分析	
29	教育学部	平 篤 志	多民族社会における少数派コミュニティの特性に関する研究	
30	教育学部	岡 田 みゆき	家庭科教育における教科書の変遷	
31	教育学部	安 東 恭一郎	研究	
32	教育学部	松 下 文 夫	マルチメディアの教育効果	
33	教育学部	小 方 朋 子	障害児教育における地域ネットワーク作り	
34	教育学部	高 倉 良 一	法学教育に関する情報収集	
35	教育学部	石 川 雄 一	学内LANへのPPP接続	
36	教育学部	小 椋 博	社会化の理論と余暇研究	
37	教育学部	時 岡 晴 美	生活様式の継承と変化に関する実証的研究	
38	教育学部	佐々木 信 行	化学データベース作成	
39	教育学部	服 部 兼 敏	心理統計の授業で使用する	
40	教育学部	山 田 貴 志	研究	
41	教育学部	阪 根 健 二	生徒指導研究	

	所 属	氏 名	研 究 課 題	備 考
42	教育学部	山下 智恵子	家庭科授業ファイルの講義室での活用	
43	教育学部	高井 忠 昌	原子レベルのコンピュータシミュレーション	
44	教育学部	藤田 和 憲	情報交換と数学教育研究	
45	教育学部	深石 博 夫	情報交換及び数学の研究	
46	教育学部	安西 一 夫	情報交換と数学教育研究	
47	教育学部	高橋 尚 志	Photoemission spectrum dataの解析と理論計算	
48	教育学部	櫻井 佳 樹	講義室等でのパソコンの利用	
49	附属坂出小学校	西浦 雅 弘	情報教育におけるインターネットの研究	
50	附属幼稚園	塩田 知 子	インターネットと幼稚園教育	
51	附属幼稚園	高井 忠 昌	幼児教育におけるインターネットの利用	
52	法学部	池端 忠 司	インターネットをはじめとするマルチメディアの法制度の研究	
53	法学部	尾崎 壽 子	データ処理	
54	法学部	石橋 可奈美	国際環境法の研究	
55	法学部	松本 タ ミ	法学における情報処理	
56	法学部	神江 伸 介	選挙研究	
57	法学部	木下 麻奈子	法学における情報処理	
58	法学部	松井 康 浩	地域国際化に関する動向	
59	法学部	湯山 智 之	国連国際法委員会の国家責任条文草案の起草作業に関する情報・研究の検索の為のインターネット閲覧	
60	法学部	松尾 邦 之	法律関係情報の収集	
61	法学部	石井 一 也	開発論・平和研究	
62	法学部	肥塚 肇 雄	金融取引の適正化の法理	
63	経済学部	平井 健 之	Eメールの利用	
64	経済学部	桑原 正 行	会計情報検索のため	
65	経済学部	西尾 富美子	地域社会分析	
66	経済学部	渡邊 英 夫	フランス現代文化研究のためのEmailとおしての資料収集	
67	経済学部	久松 博 之	計量経済モデルの推定と検定	
68	経済学部	藤井 宏 史	マクロ経済のシミュレーション	
69	経済学部	宮脇 秀 貴	現代企業の会計情報システム分析	
70	経済学部	曾 道 智	コンフリクトのゲーム理論解析	
71	経済学部	本田 道 夫	知的情報処理の研究	
72	経済学部	岡田 徹太郎	経済政策	
73	経済学部	斉藤 和 也	西洋古典学関係の情報収集	
74	経済学部	大野 拓 行	日本経済の分析	
75	経済学部	大野 拓 行	計量経済分析	
76	経済学部	長山 貴 之	日本の予算に関する実証分析	
77	経済学部	井上 貴 照	国際経済分析	
78	経済学部	尾崎 万 枝	経営分析	
79	経済学部	星野 良 明	経済理論の研究	
80	経済学部	板倉 宏 昭	研究	
81	経済学部	安藤 博 子	UNIXの利用	
82	経済学部	三谷 宗 子	UNIXシステム管理	
83	経済学部	増井 久 之	情報リテラシー、経営情報システム、電子政府、電子商取引、ビジネスモデル	
84	経済学部	喜田 恵津子	財務分析	

	所 属	氏 名	研 究 課 題	備 考
85	経済学部	木 村 住 枝	経済分析	
86	経済学部	崔 康 植	研究	
87	経済学部	細 川 滋	経済分析	
88	経済学部	金 澤 理恵子	教育のサポート	
89	経済学部	井 上 信 一	日系企業に於ける管理会計の実態	
90	経済学部	井 上 信 一	日系企業の原価管理システムの研究	
91	経済学部	井 上 信 一	グローバル企業の原価管理システムの研究	
92	経済学部	井 上 信 一	在英日系企業の原価管理	
93	経済学部	井 上 信 一	在英日系企業の管理会計の調査	
94	経済学部	井 上 信 一	利益管理情報と原価管理情報に関する調査研究	
95	経済学部	植 木 英 治	Eメール、掲示板	
96	経済学部	佐 藤 幹 子	日本経済分析	
97	経済学部	高 塚 創	地域成長の数値解析	
98	経済学部	梶 脇 裕 二	研究	
99	経済学部	松 岡 久 美	経営関連情報の検索のため	
100	経済学部	RALPH DEGEN	授業用及び研究資料のオンライン検索	
101	経済学部	三 原 麗 珠	人文・社会・自然科学の研究	
102	経済学部	姚 峰	経済モデル解析	
103	経済学部	中 村 邦 彦	情報処理技法の研究	
104	経済学部	高 木 文 夫	研究	
105	経済学部	佐 藤 公 一	西洋思想関係の情報検索	
106	経済学部	佐 藤 保 久	研究及び情報収集・検索	
107	経済学部	原 直 行	経済研究	
108	経済学部	関 義 雄	設計品質に関する研究	
109	経済学部	大 賀 睦 夫	E-mail, NetWareの利用	
110	経済学部	雲 和 広	地域経済成長と地域間生産要素移動に関する研究	
111	経済学部	岡 田 ツヤ江	経営分析	
112	経済学部	山 下 隆 資	情報収集	
113	経済学部	藤 原 敦 志	金融論の検証	
114	経済学部	守 矢 信 明	フランス文化圏の資料収集	
115	経済学部	山 口 博 幸	企業行動の多変量解析	
116	経済学部	原 真 志	産業集積に関する研究	
117	経済学部	小 宮 一 高	研究にともなう資料収集のため	
118	経済学部	最 上 英 明	電子メールの利用	
119	経済学部	清 水 真 志	政治経済学(経済原論)の研究	
120	経済学部	佐 藤 忍	国際労働力移動の研究	
121	経済学部	LRONG. LIM	経営・統計分析	
122	経済学部	ラナデ. R. R	数理経済学	
123	経済学部	藤 村 和 宏	研究(サービス・マーケティングの研究)	
124	経済学部	安 井 敏 晃	米国の保険学研究誌の論文、記事入手	
125	経済学部	朴 鏡 杓	管理会計研究	
126	経済学部	金 東 吉	経営資料検索	
127	経済学部	宍 戸 榮 徳	オペレーションズ・リサーチの応用	
128	経済学部	渡 辺 己	データ処理	
129	経済学部	井 原 理 代	財務会計の研究	

	所 属	氏 名	研 究 課 題	備 考
130	経済学部	堀 啓 造	マーケティングの心理的研究	
131	経済学部	瀧 川 一 幸	ドイツ語教材, ドイツ語教育法のための研究	
132	経済学部	稲 田 道 彦	地域文化研究	
133	工学部	澤 田 秀 之	研究	
134	工学部	角 田 直 人	高分解能電子顕微鏡像の計算機シミュレーション	
135	工学部	郭 書 祥	マイクロ能動ガイドワイヤカテーテルシステム	
136	工学部	仲 谷 英 夫	地球科学データベース構築	
137	工学部	中 西 俊 介	非線型光学効果の数値計算	
138	工学部	秦 清 治	学外からメールサーバへのアクセスのため	
139	工学部	吉 野 文 雄	研究	
140	工学部	寺 林 優	地球科学データベース構築	
141	工学部	香 川 考 司	研究	
142	工学部	生 越 重 章	モバイルマルチメディアの研究	
143	工学部	加 藤 大志朗	画像処理・ネットワークソフトに関する研究	
144	工学部	堀 川 洋	パターン認識に関する研究	
145	工学部	守 田 秀 則	土地利用モデルの開発	
146	工学部	高 津 義 典	研究	
147	工学部	古 川 善 吾	ソフトウェアの分散開発方式の研究	
148	工学部	山 崎 敏 範	パターン情報処理	
149	工学部	山 崎 敏 範	感性情報処理の研究	
150	工学部	和 田 隆 広	研究	
151	工学部	服 部 哲 郎	パターン情報の処理と認識に関する研究	
152	工学部	呉 景 龍	研究	
153	工学部	吉 田 秀 典	大規模地下空洞掘前解析とその評価	
154	工学部	石 井 知 彦	スピン分極相対論電子状態の解析	
155	工学部	松 原 行 宏	感性工学と製品開発に関する基礎的研究	
156	工学部	今 井 慈 郎	計算機ネットワーク環境の実証的研究	
157	工学部	今 井 慈 郎	ネットワークセキュリティに関する基礎的研究	
158	工学部	FINN, William Daniel	Foundation Geodyuamics	
159	工学部	石 川 浩	信頼性工学に関する研究	
160	工学部	井 面 仁 志	学外からのメールサーバへの接続	
161	工学部	井 面 仁 志	学外からのメールサーバへの接続	
162	工学部	白 木 渡	学外からのメールサーバへの接続	
163	工学部	河 原 能 久	研究	
164	工学部	神 崎 正	研究	
165	工学部	井 筒 勝 彦	研究	
166	工学部	角 道 弘 文	研究	
167	工学部	野々村 敦 子	研究	
168	工学部	青 柳 省 吾	研究	
169	工学部	堺 孝 司	研究	
170	工学部	野 田 茂	研究	
171	工学部	松 島 学	研究	
172	工学部	長谷川 修 一	研究	
173	工学部	山 中 稔	研究	

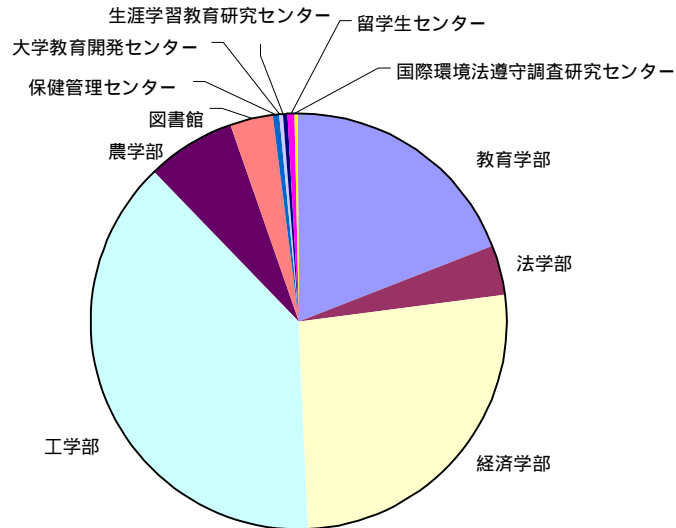
	所 属	氏 名	研 究 課 題	備 考
174	工学部	増 田 拓 朗	研究	
175	工学部	土 井 健 司	研究	
176	工学部	森 下 一 男	研究	
177	工学部	守 屋 均	研究	
178	工学部	藤 田 晃	研究	
179	工学部	垂 水 浩 幸	研究	
180	工学部	北 島 博 之	研究	
181	工学部	安 藤 一 秋	研究	
182	工学部	清 水 秀 明	研究	
183	工学部	岡 本 研 正	研究	
184	工学部	最 所 圭 三	研究	
185	工学部	中 川 清	研究	
186	工学部	丹 治 裕 一	研究	
187	工学部	筑 瀬 靖 子	研究	
188	工学部	尹 禮 分	研究	
189	工学部	Vatchkov Gantcho	研究	
190	工学部	石 井 明	研究	
191	工学部	山 口 順 一	研究	
192	工学部	石 丸 伊知郎	研究	
193	工学部	高 橋 悟	研究	
194	工学部	林 純一郎	研究	
195	工学部	高 本 喜 一	研究	
196	工学部	石 原 秀 則	研究	
197	工学部	能 見 公 博	研究	
198	工学部	三 原 豊	研究	
199	工学部	大 平 文 和	研究	
200	工学部	平 田 英 之	研究	
201	工学部	橋 口 原	研究	
202	工学部	大 上 祐 司	研究	
203	工学部	細 木 真 保	研究	
204	工学部	江 原 隆一郎	研究	
205	工学部	小 川 一 文	研究	
206	工学部	若 林 利 明	研究	
207	工学部	掛 川 寿 夫	研究	
208	工学部	国 重 和 俊	研究	
209	工学部	馮 旗	研究	
210	工学部	小 柴 俊	研究	
211	工学部	品 川 一 成	研究	
212	工学部	田 中 克 志	研究	
213	工学部	江 島 正 毅	研究	
214	工学部	神 垣 良 昭	研究	
215	工学部	伊 藤 寛	研究	
216	工学部	須 崎 嘉 文	研究	
217	工学部	清 田 雄 彦	研究	
218	工学部	小 松 孝 二	研究	

	所 属	氏 名	研 究 課 題	備 考
219	工学部	塚 本 一 義	研究	
220	工学部	川 口 政 秀	実験実習補助のため	
221	工学部	松 本 直 通	実験実習補助のため	
222	工学部	吉 田 俊 一	実験実習補助のため	
223	工学部	立和名 慎 一	実験実習補助のため	
224	工学部	絹 田 志 穂	実験実習補助のため	
225	工学部	鈴 木 崇 彦	実験実習補助のため	
226	工学部	俵 原 基 吉	事務処理支援業務	
227	工学部	吉 村 英 徳	学外からメールサーバへのアクセスのため	
228	工学部	富 永 浩 之	情報科学の教育研究	
229	工学部	荒 川 雅 生	研究	
230	工学部	末 永 慶 寛	海域環境変化の数値シミュレーション	
231	工学部	中 山 千 晶	研究	
232	工学部	岩 部 達 雄	研究	
233	工学部	岩 本 直 樹	研究	
234	工学部	鶴 町 徳 昭	研究	
235	工学部	河内山 隆 紀	計測データ解析	
236	農学部	石 田 智 之	農地工学における数値計算	
237	農学部	永 瀬 雅 啓	生体防御機構に関する研究	
238	農学部	片 岡 郁 雄	果樹生産分野における研究打ち合わせと資料収集	
239	農学部	高 村 武二郎	園芸植物の種苗生産に関する研究	
240	農学部	古 本 敏 夫	情報検索等	
241	農学部	山 本 弘 幸	病害抵抗性の発現機構に関する研究	
242	農学部	佐 藤 正 資	生理活性試験結果のデータ処理	
243	農学部	秋 光 和 也	e-mail	
244	農学部	早 川 茂	食品蛋白質の生理的・加工的特性の解析	
245	農学部	麻 田 恭 彦	担子菌類の分子生物学	
246	農学部	竹 川 薫	分裂酵母の塩基配列及び遺伝子構造の解析	
247	農学部	別 府 賢 治	果樹生産分野における資料収集	
248	農学部	豊 田 正 範	作物学における統計解析	
249	農学部	亀 山 宏	環境政策シュミレーションCGEモデルの構築	
250	農学部	多 田 邦 尚	E-Mailによるパソコン通信,海洋環境データ処理	
251	農学部	加 藤 尚	研究	
252	農学部	彌 永 孝 一	園芸学実験実習・パソコン入門	
253	農学部	鈴 木 晴 雄	マルチの温度効果	
254	農学部	山 内 高 圓	鶏腸管の研究	
255	附属図書館	鈴 木 美智子	メディアプラザPC利用・管理	
256	附属図書館	鶴 川 多歌子	他機関のデータベース等利用・メディアプラザ管理	
257	附属図書館	岩 澤 尚 子	学生用PCの管理	
258	附属図書館	藤 谷 栄	図書受入目録業務	
259	附属図書館	山 本 博 子	メディアプラザPC利用・管理	
260	附属図書館	片 山 恒 信	図書館システムのメンテナンス	
261	附属図書館	龍 満 馨	メディアプラザPC利用・管理	
262	附属図書館	竹 島 雅 美	メディアプラザのメンテナンス	
263	附属図書館	櫛 橋 一 雅	図書館システム管理 メディアプラザ管理	

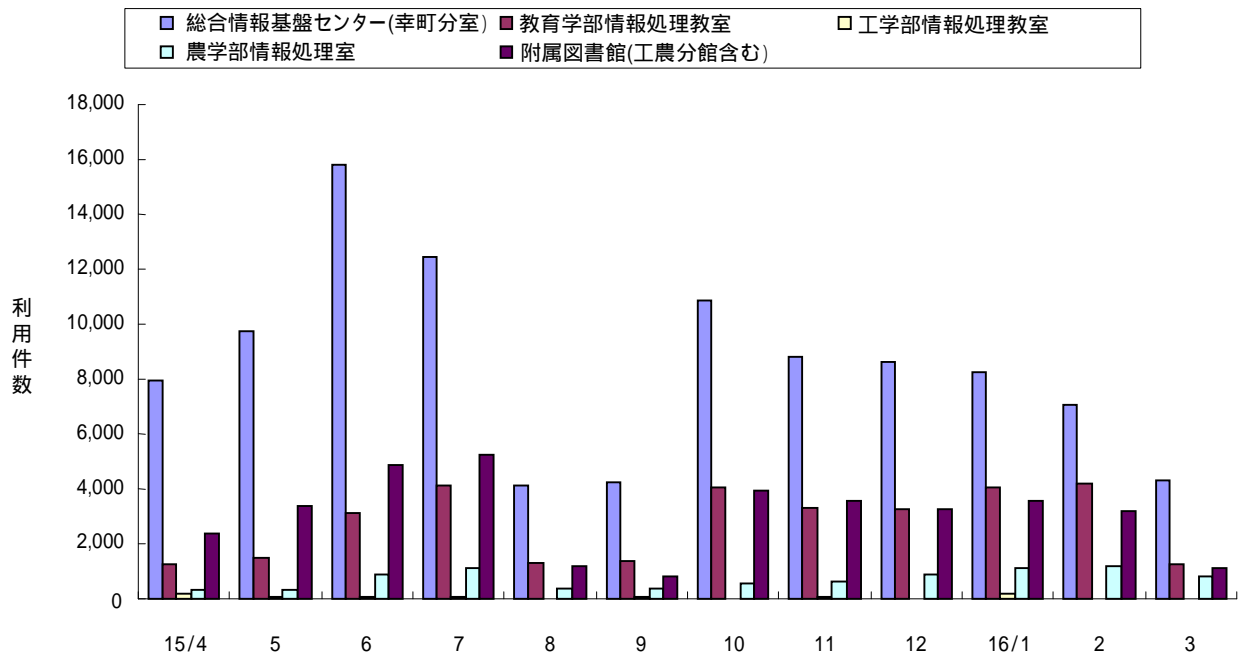
	所 属	氏 名	研 究 課 題	備 考
264	保健管理センター	鎌 野 寛	学外データベース利用、研究データ処理	
265	大学教育開発センター	稲 永 由 紀	研究のため	
266	生涯学習教育研究センター	清 國 祐 二	教養科目の教材研究	
267	留学生センター	塩 井 実 香	日本語学・日本語教育に関する調査・研究	
268	国際環境法遵守調査研究センター	石 橋 可奈美	国際環境法の遵守調査研究	

計算機利用統計

平成15年度利用申請



平成15年度パソコン室利用状況



平成15年度パソコン室利用統計

総合情報基盤センター幸町分室

利用件数

	15/4	5	6	7	8	9	10	11	12	16/1	2	3	計
教育学部	219	192	362	273	233	184	186	187	179	202	209	126	2,552
法学部	1,494	1,874	2,949	2,273	973	1,243	3,287	2,398	2,323	2,445	1,922	1,276	24,457
経済学部	5,669	7,500	11,979	9,388	2,149	2,323	6,899	5,914	5,832	5,253	4,583	2,587	70,076
工学部	31	37	85	106	114	111	116	127	75	79	125	102	1,108
農学部	82	75	140	220	349	221	130	148	138	105	172	176	1,956
その他	428	87	290	189	291	188	239	57	81	196	46	65	2,157
計	7,923	9,765	15,805	12,449	4,109	4,270	10,857	8,831	8,628	8,280	7,057	4,332	102,306

延べ利用時間(分)

	15/4	5	6	7	8	9	10	11	12	16/1	2	3	計
教育学部	12,392	13,927	22,961	18,695	17,217	13,537	9,760	10,793	13,177	14,458	16,974	7,548	171,439
法学部	74,360	106,812	175,333	150,758	61,909	85,658	155,164	147,085	160,641	174,420	155,152	91,535	1,538,827
経済学部	332,249	530,655	902,891	678,991	153,590	177,034	365,544	403,453	412,436	372,631	347,723	211,871	4,889,068
工学部	1,033	2,034	4,234	7,464	10,518	7,566	8,278	8,774	4,952	5,405	9,639	8,547	78,444
農学部	2,365	3,611	6,740	21,706	24,975	21,372	10,737	12,789	16,444	9,839	16,030	15,188	161,796
その他	17,437	5,757	5,557	4,728	7,141	3,537	3,519	2,727	3,508	3,003	2,697	2,648	62,259
計	439,836	662,796	1,117,716	882,342	275,350	308,704	553,002	585,621	611,158	579,756	548,215	337,337	6,901,833

教育学部情報処理教室

利用件数

	15/4	5	6	7	8	9	10	11	12	16/1	2	3	計
教育学部	638	1,147	2,158	2,963	894	1,036	3,240	2,609	2,482	3,054	3,302	885	24,408
法学部	0	5	19	84	21	18	115	113	152	238	169	91	1,025
経済学部	287	339	859	939	116	150	588	478	486	550	493	165	5,450
工学部	1	16	46	81	29	61	44	56	51	117	153	71	726
農学部	128	2	37	51	39	25	35	23	27	14	36	15	432
その他	204	5	13	26	206	103	56	25	38	95	36	7	814
計	1,258	1,514	3,132	4,144	1,305	1,393	4,078	3,304	3,236	4,068	4,189	1,234	32,855

延べ利用時間(分)

	15/4	5	6	7	8	9	10	11	12	16/1	2	3	計
教育学部	47,580	93,912	146,674	223,000	77,988	83,807	172,691	183,478	197,570	250,716	302,473	66,658	1,846,547
法学部	0	391	1,423	6,466	1,467	1,027	6,075	12,002	10,681	19,432	19,705	10,530	89,199
経済学部	18,283	25,671	70,442	69,349	9,465	10,437	29,019	33,417	33,198	43,673	41,237	14,141	398,332
工学部	38	670	2,002	4,401	2,476	3,528	2,249	4,551	4,567	10,343	13,392	6,497	54,714
農学部	1,611	69	1,770	3,643	3,215	1,598	2,161	1,469	1,712	1,152	3,711	1,206	23,317
その他	1,026	79	874	1,896	4,454	2,346	2,744	2,214	3,128	3,146	3,284	574	25,765
計	68,538	120,792	223,185	308,755	99,065	102,743	214,939	237,131	250,856	328,462	383,802	99,606	2,437,874

工学部情報処理教室

利用件数

	15/4	5	6	7	8	9	10	11	12	16/1	2	3	計
教育学部	0	0	0	0	0	0	0	0	0	0	0	0	0
法学部	0	0	0	0	0	0	0	0	0	0	0	0	0
経済学部	0	0	0	0	0	0	0	0	0	0	0	0	0
工学部	98	50	60	27	0	0	19	36	14	39	4	1	348
農学部	2	0	0	0	0	0	0	0	0	0	0	0	2
その他	93	13	25	6	0	85	10	0	6	160	0	0	398
計	193	63	85	33	0	85	29	36	20	199	4	1	748

延べ利用時間(分)

	15/4	5	6	7	8	9	10	11	12	16/1	2	3	計
教育学部	0	0	0	0	0	0	0	0	0	0	0	0	0
法学部	0	0	0	0	0	0	0	0	0	0	0	0	0
経済学部	0	0	0	0	0	0	0	0	0	0	0	0	0
工学部	4,352	4,085	5,470	2,852	0	0	845	2,534	740	2,220	22	1	23,121
農学部	11	0	0	0	0	0	0	0	0	0	0	0	11
その他	243	453	1,617	677	0	71	23	0	222	132	0	0	3,438
計	4,606	4,538	7,087	3,529	0	71	868	2,534	962	2,352	22	1	26,570

農学部情報処理室

利用件数

	15/4	5	6	7	8	9	10	11	12	16/1	2	3	計
教育学部	0	0	0	0	0	0	5	3	4	5	0	0	17
法学部	0	0	0	1	2	0	2	2	2	0	3	2	14
経済学部	0	0	4	0	0	0	0	1	2	0	1	0	8
工学部	0	0	1	0	0	0	2	1	0	1	0	2	7
農学部	252	286	828	1,080	300	307	523	617	848	1,029	1,142	786	7,998
その他	61	3	65	20	58	58	13	15	13	64	11	9	390
計	313	289	898	1,101	360	365	545	639	869	1,099	1,157	799	8,434

延べ利用時間(分)

	15/4	5	6	7	8	9	10	11	12	16/1	2	3	計
教育学部	0	0	0	0	0	0	314	226	223	278	0	0	1,041
法学部	0	0	0	14	54	0	33	80	102	0	218	35	536
経済学部	0	0	568	0	0	0	0	54	338	0	8	0	968
工学部	0	0	108	0	0	0	157	86	0	9	0	63	423
農学部	30,303	23,235	54,661	104,701	22,659	25,323	31,913	47,740	61,970	79,893	99,466	67,718	649,582
その他	378	162	942	542	280	356	397	781	423	619	463	379	5,722
計	30,681	23,397	56,279	105,257	22,993	25,679	32,814	48,967	63,056	80,799	100,155	68,195	658,272

附属図書館(工学部及び農学部分館含む)

利用件数

	15/4	5	6	7	8	9	10	11	12	16/1	2	3	計
教育学部	377	532	674	690	208	86	535	516	394	500	495	165	5,172
法学部	450	654	858	848	167	140	978	734	633	741	742	279	7,224
経済学部	919	1,642	2,483	2,743	436	249	1,934	1,767	1,756	1,775	1,441	433	17,578
工学部	217	387	730	644	130	89	475	784	657	779	567	155	5,614
農学部	715	1,121	1,421	1,579	578	539	1,219	1,084	1,117	1,140	1,228	873	12,614
その他	607	231	223	278	204	182	233	242	167	252	212	170	3,001
計	3,285	4,567	6,389	6,782	1,723	1,285	5,374	5,127	4,724	5,187	4,685	2,075	51,203

延べ利用時間(分)

	15/4	5	6	7	8	9	10	11	12	16/1	2	3	計
教育学部	16,256	24,847	31,816	33,476	10,526	4,190	20,575	26,401	20,456	26,825	26,983	10,168	252,519
法学部	16,762	28,845	40,573	37,746	6,099	6,697	35,866	32,999	29,650	35,622	30,237	11,695	312,791
経済学部	37,877	85,344	129,155	130,905	19,857	12,237	85,665	97,445	98,319	95,906	67,595	20,241	880,546
工学部	7,097	13,651	26,791	22,948	4,490	5,925	21,675	27,313	25,326	28,716	18,765	6,731	209,428
農学部	32,347	47,610	60,065	73,841	31,875	27,555	60,302	65,118	62,167	71,541	72,833	51,412	656,666
その他	25,351	12,650	15,254	19,523	10,199	12,526	15,804	19,287	13,829	17,502	15,899	15,011	192,835
計	135,690	212,947	303,654	318,439	83,046	69,130	239,887	268,563	249,747	276,112	232,312	115,258	2,504,785

平成15年度総合情報基盤センターPC利用状況

PCルーム1, 2, 3

		月	火	水	木	金
8:50	前					マルチメディア処理
	後					
10:30	前	経営財務論	マルチメディア処理 調査データ分析	空間計量分析B 統計学B 特殊講義		
	後				経済統計	経済統計
13:00	前		情報処理入門イ	情報処理入門ハ	経営財務論 (特)プログラミング 都市・地域分析 F/W	教養ゼミナール
	後					
14:40	前	演習	演習 情報処理入門口		情報法演習 法社会学演習	
	後	演習	演習			
16:20	前	情報処理基礎 計算機基礎	情報処理基礎		演習	演習
	後	人間環境教育基礎 演習			演習	演習
18:00	前			情報システム	情報法 産業情報システム論	経営管理論 B
	後					情報科学
19:40	前			情報システム	情報法 産業情報システム論	経営管理論 B
	後					情報科学
備考		集中講義等： 新入生情報処理ガイダンス(法学部)，生涯学習教育研究センター公開講座(インターネット時代の証券投資： 賢い投資の冒険，電子メール・デジタルカメラ入門講座)，法科大学院教育システム研修				

平成15年度総合情報基盤センターPC利用状況

教育学部情報処理教室1, 2

		月	火	水	木	金
8:50	前					
	後			マルチメディアリテラシ(H)		
10:20						
10:30	前	数学科教育論		マルチメディアリテラシ(A)		
	後					
12:00						
13:00	前		数学情報論			設計製図
	後					
14:30						
14:40	前					
	後		マルチメディアリテラシ(D)		マルチメディアリテラシ(C)	
16:10						
16:20	前	情報処理基礎	マルチメディアリテラシ(B)			
	後	マルチメディアリテラシ(I)				
17:50						
18:00	前					
	後					
19:30						
19:40	前					
	後					
21:10						
備考		集中講義等：生活情報処理				

注意：

農学部及び工学部については運用を任せており統計はなし。

教育学部情報処理教室については申し出のあったもののみである。

この表に記載しているのは授業で使ったもののみで、空き時間には学生に自由に使用させている。

平成15年度総合情報基盤センターPC利用状況

医学部マルチメディア実習室

前期	月曜日	火曜日	水曜日	木曜日	金曜日
8:50 -10:20					
10:30 -12:00				情報科学	
13:00 -14:30					データとデータ解析
14:40 -16:10		生命科学			
18:30 -21:30					大学院地域看護学演習

後期	月曜日	火曜日	水曜日	木曜日	金曜日
8:50 -10:20					
10:30 -12:00		情報科学			
13:00 -14:30		情報科学			
14:40 -16:10		情報科学			

年報 第2号

2005年2月発行

編集 香川大学総合情報基盤センター

発行 香川大学総合情報基盤センター

〒761-0793 木田郡三木町池戸1750-1

087-891-2266 (ダイヤルイン)

〒760-8523 高松市幸町2-1

087-832-1292 (ダイヤルイン)
