

工学系学生のための情報処理集合教育環境の設計と構築

中山 仁† 大西 淑 雅†
末 永 正†† 有田 五次郎†††

ワークステーションやパーソナルコンピュータ等の小型計算機システムの高性能化に伴い、近年それらを中心とする分散型のシステムが、多くの分野で大型の汎用機を中心とするシステムにとって代わりつつある。大学における一般情報処理教育の環境も例外ではなく、今後の計算機利用形態の高度化を考えると、専用端末での TSS という形態は機能面、性能面共に不十分であろう。九州工業大学では今回、ワークステーションを中心とする分散型システム構成の集合教育システムを構築し、ほぼ満足すべき結果を得ることができた。本稿では、このシステムの設計と実現を通して、工学系学生に必要な情報処理教育環境の条件、および分散型システム構成により集合教育環境を構築する上での運用面での課題と、その解決法について述べる。特に性能面については、運用中のシステム各部のログングデータを分析することにより、今回の実現方法の効果を確認した。

A Design and Construction of Educational Computer Environment for Engineering Students

HITOSHI NAKAYAMA,† YOSHIMASA OHNISHI,† TADASHI SUENAGA††
and ITSUJIRO ARITA†††

In recent years, distributed computer systems that are constructed with workstations and personal-computers become superior to mainframe systems in many aspects including performance and human-interface. Educational systems in universities become also to require high-level system features that time-sharing systems on mainframes cannot provide. In this paper, we discuss the requirements for educational computer system especially for engineering students and describe the problem of constructing and managing such a large distributed educational system, using our experience of the design and the construction of a distributed system in Kyushu Institute of Technology.

1. はじめに

情報処理教育の重要性が叫ばれて久しい。近年では、コンピュータリテラシーのための教育が特に注目され、理工系の学生のためばかりでなく、いわゆる文科系学生のための一般情報処理教育も行われるようになった¹⁾。

一口に一般情報処理教育といっても、対象学生の専門分野によって教育内容が違うのは当然であり、そのための必要な計算機システムも異なってくるはずであ

る。しかしながら、実際の教育環境は、情報処理教育センターのような学生施設を共同で利用する形態となっていることが多く、利用時間やシステム資源の制約などによって、必要な教育を施せないことにもなりかねない。

大学における情報処理教育のためのシステム環境に関する報告^{2)~4)}は多数あるが、これらのシステムを構築するに至った理由を教育的側面、性能的側面、および運用管理（コスト）的側面から総合的に論じたものは少ない。

本稿では、九州工業大学情報科学センターにおける情報処理教育用システムを一つの事例として、工学系大学における教育用システム環境のありかたと、その具体的な構築結果について述べる。以下、まず第2章では、工学系学生に対する情報処理教育環境の特性とその必要条件、さらに現状での問題点について考察する。次に3章と4章では、九州工業大学のシステム事例における基本的な設計方針および具体的な設計をそ

† 九州工業大学情報科学センター
Information Science Center of Kyushu Institute of Technology

†† 近畿大学九州工学部
Faculty of Engineering, Kinki University in Kyushu

††† 九州工業大学情報工学部
Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology

れぞれ紹介する。なお設計に際して、システムの小規模なモデルを設定し、その上で学生演習を想定したいくつかの実験を行うことにより、システムの性能に関する事前評価を行った。4章ではこの実験と評価結果についても述べる。また、運用を開始したシステムにおいて、4章と同様な性能の測定および最大負荷時のシステム各部の稼働状況の測定を行い、設計の事後評価を行った。5章ではその結果について述べる。また、システムの問題点とその対策についても議論する。

2. 教育環境としての必要条件

工学系学生に対する一般情報処理教育は、システム構築能力の育成を目的とし⁹⁾、コンピュータを単なる道具として使用するだけでなく、内部的な処理方法に関する知識もある程度修得できる内容であることが望ましい。具体的な内容として、文献5)では、

- 使用する計算モデルで問題を表現するための抽象化
- モデル化された問題を解決するアルゴリズムとその性能評価
- 大規模な問題を解決するためのモジュール化

などについての教育が挙げられている。このような教育目標を達成するためには、

- (1) 充実した教育用システム
- (2) カリキュラム上の十分な教育時間
- (3) 少人数クラスでの行き届いた教育体制

が必要である。しかしながらわが国の大学においては、これらすべてを満足のいく形で実現できているところは少ない。特に(2)および(3)については、大学運営上の構造的な問題が含まれており、解決に長期間を要するであろう。そこで本稿では、技術的に改善の余地がある(1)について考察する。

現時点において、情報処理教育のためのシステム環境の設計には、本来の教育内容に適したシステムをめざすという面だけでなく、以下の点も考慮する必要がある。

(a) 情報科学の授業担当者数、予算、教室数、カリキュラム上の時間数などの制約から、50名から100名程度の集合教育を行わざるをえないこと。

(b) 情報処理教育の授業だけに限らず、インフラストラクチャとして計算機を広く利用できること。

(c) システムの管理運用に、人的資源を含めて多くのコストをかけられないこと。

(d) 専門課程における教育研究に向けての一貫性

が保たれること。

(e) ソフトウェアツールの開発および活用を促進でき、軽快で柔軟性に富むシステム環境であること。特に(b)については、今後の情報化社会に対応していくためにも、その重要性がますます増大していくと思われる。

また、工学系の特徴として、研究上必要なツールを手作りしなければならないこともあり、ツール作りが教育研究の一環として行われることも少なくない。さらに、他人が作ったツールを評価し、改良することさえある。情報処理の分野では、このようなツールがフリーソフトウェア(フリーウェア)として多数流通しており、これらをうまく活用できる能力や、既存のツールが個々の問題解決に向かない場合に目的の適したものを開発しうる能力を身につけることも重要である。そのためには、必要な処理をメニューに従って指示するだけでことが足りるような初心者向けの環境よりも、使用上の制約が少なく柔軟で軽快な作業環境が望ましいといえる。

従来、(a)および(c)の点ばかりが強調され、メインフレームを中心とした集中型教育システムが構築されてきた。しかし、1台のメインフレームによる閉じた環境では、ネットワーク技術やグラフィックユーザインターフェース(GUI)技術に関する演習がそのメインフレーム固有の機能を前提にしたものになり、これらの基礎技術を広く応用学習しにくいという問題がある。また、情報処理教育の必要性が増し、その集合教育規模を大きくせざるを得ない現状では、一斉使用時の応答性の悪化や障害発生時の影響も無視できない。このような点から、教育環境としての再検討が必要であると思われる。

こうした集中型の計算機システムに対して、パーソナルコンピュータ(PC)やワークステーション(WS)、各種機能サーバ(計算サーバやファイルサーバなど)を高速なネットワークで結んだ分散型システムが普及してきている。特に情報を専門とする学科では、この分散型システムを使った教育研究が数年前から行われており、この傾向は情報を専門としない学科にも波及しつつある。このような状況の中で、専門課程と全く独立したシステム環境に基づく教育を行うのは好ましくない。

3. 設計の基本方針

九州工業大学情報科学センターではこれまで、おも

にメインフレームの TSS によって教育用計算機環境を提供してきた。しかし、前節で述べた教育システムとしての要件から見ると、

- 利用形態の高度化、多様化に対応していくには、能力的に不足である。
- システムが比較的閉鎖的で、学内ネットワークを通じて他のシステムと相互通信する上で制約が多い。
- フリーウェアの流通が比較的少ない。

などの問題があった。

そこで平成4年のシステム更新において、システムの構成を根本的に見直し、次のような方針に基づいて新しいシステムを構築することにした。

- 単一のメインフレームにかえて、近年急速にコストパフォーマンスの向上した高機能ワークステーションを複数設置し、システム全体としての性能の向上をめざす。
- オペレーティングシステム (OS) としては、工学系 (特に情報系) 分野における流通ソフトウェアの豊富さ、およびそれらの移植性や利用上のノウハウの共有などを第一義的に考え、またシステムメンテナンスの容易さも考慮して、全世界的に普及している UNIX 準拠のものを採用する。
- ネットワーク機能についても、ネットワーク接続機器の豊富さ、既存の学内ネットワークや広域研究ネットワークとの相互接続性、およびネットワーク関連ソフトウェアの互換性などから、TCP/IP をベースとしたものとする。
- 文字情報ばかりでなく画像情報も容易に扱えるように、高度な GUI 機能を備えたウィンドウシステムを導入する。
- 工学系、とくに情報系の学生が多く利用することを考慮し、システムには過度に利用者インターフェースの殻をかぶせることはせず、できるだけ「生」のシステムが見えるようにする。このことは、システムをある目的のための単なる道具として使用する場合に、余分な周辺知識も必要となるため初心者向きでなくなる可能性がある。しかし、そのことよりもシステム利用上の柔軟さ、および軽快さを重視する。
- システム利用上の制限は、利用形態の多様性への足かせにならないよう、必要最小限にとどめる。
- 大学に所属するすべての職員、学生 (あわせて約7000名) に利用権を与え、講義、演習目的にとどまらず広範な計算機環境を提供する。

また集合教育環境としての必要性から、

- 最高で約400名の学生が、それぞれ一端末を占有して同時に演習できる。
 - 利用するワークステーション、端末によって、利用者環境に差異が出ない。
- などの条件を加えた。

一方管理運用の見地からは、システム障害への対策として、

- 主要なシステム資源はできるだけ二重化するか、あるいは冗長に配置する。
 - 障害時のバックアップシステムへの切り替えは、できる限り自動化する。
 - サブシステムをできるだけ均一化し、管理手順が標準化できるようにする。
- などを盛り込んだ。

さらに定常的な保守管理作業を省力化するため、システム情報の収集やシステムの制御は、できるだけ集中化、自動化することとし、そのために必要なユーティリティの整備などを行っていくことにした。

4. 詳細設計

4.1 全体の構成

図1に新システムと学内ネットワークシステムの概要を示す。九州工業大学は戸畑地区 (工学部) と飯塚地区 (情報工学部) の二つのキャンパスを持っている。これらのキャンパスは約40km離れており、それぞれに学内ネットワークが整備されている。各キャンパスネットワークには多くの学科 (飯塚地区では全学科) の学科内ネットワークが接続し、通信を行うことができる。また両地区のネットワークは高速デジタル回線を用いて結ばれ、さらに WIDE (Widely Integrated Distributed Environments) および KARRN (Kyushu Area Regional Research Network) の二つの広域研究ネットワークにも接続している。また、現在は孤立したシステムとなっている、図書館の文献検索システムや大学事務の情報処理システムも、学内ネットワークに接続し相互運用を可能にするための準備を行っている^{6),7)}。

さて、従来のセンターシステム (メインフレーム) もこのネットワークと接続していたが、接点は一か所だけで、また利用上の制約の多いものであった。新システムは複数のサブシステムによって構成し、それぞれが学内ネットワークに接続する、という形態をとる。

サブシステムは教育、研究、管理の三種類とする。教育サブシステムは本システムの中核部分であり、これまで述べてきたような教育用計算機環境を実現する。教育サブシステムは両地区に同等のものをそれぞれ用意し、利用者は自分の所属する地区のシステムを使用する。

研究サブシステムは、主に研究利用登録を行った利用者に特殊な装置を提供するもので、例えば数値計算プロセッサ（ミニスーパーコンピュータ）のような計算サーバや、XYプロッタ、高品位カラープリンタなどの入出力機器などを備える。研究システムの内容はそれぞれの地区で異なっているが、利用者はネットワークを通じて別の地区のシステムを利用することも可能である。

管理サブシステムでは、システム全体の監視および制御を行うほか、ライブラリ、ユーティリティの開発や、フリーウェアの導入、移植なども行う。センターの人員が両地区に分散しているため、管理システムも教育システムと同様に同等なものを地区ごとに設置する。

以下の各節では教育サブシステムについて、その設計の詳細を述べる。なお、研究サブシステムおよび管

理サブシステムの詳細については本稿の主旨からはずれるので割愛する。

4.2 機器構成

基本方針により、ワークステーションを中心としたシステム構成にすることは決定したものの、その具体的な構成については二つの方式を考えることができる。一つは、利用者端末を含めシステム全体をワークステーションで構成する方式、そしてもう一つは、利用者端末をXウィンドウ端末（以下X端末）とし、複数の端末に対して1台のワークステーションをホストとして置くやり方である。

ワークステーションのみで構成する方式の利点は、その性能面にある。最下位機種のワークステーションであっても、画面表示などの応答性はX端末のそれを上回る。また、システム全体の利用者数に関わらず、1台のワークステーションは1人の利用者だけにサービスすればよいので、多人数で利用する場合にもシステム性能の低下などを心配する必要はない。一方X端末方式の場合には当然、1台のワークステーションが複数台のX端末（すなわち複数の利用者）に対してサービスするという形態をとることになる。したがって、ホスト役のワークステーションの能力としては一

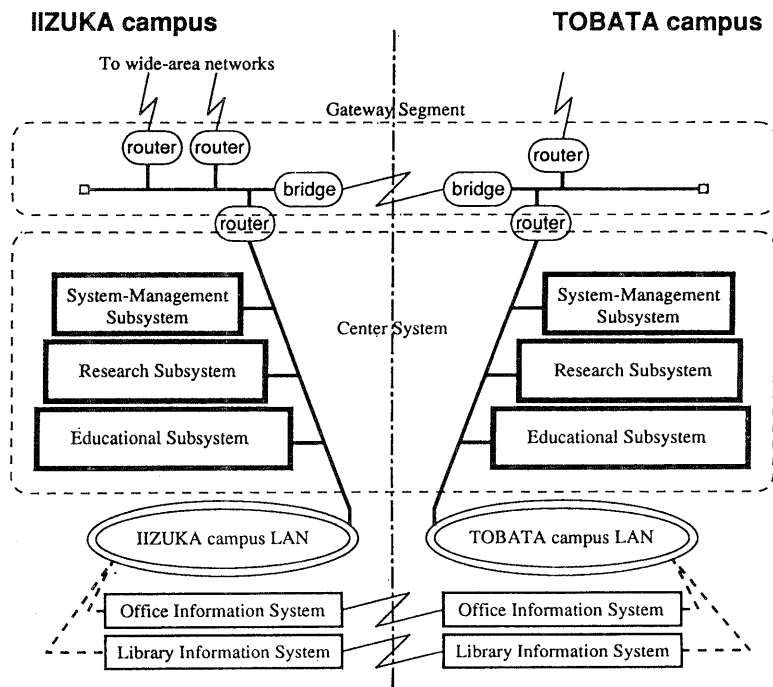


図1 システム構成の概要

Fig. 1 Outline of system configuration.

定水準以上のものが必要となるし、1台のワークステーションが何台の端末にサービスできるか、についても検討しなければならない。本来少人数での利用を想定して設計されているワークステーションを多くの利用者で使用する場合、十分な性能が確保できるかどうかも心配である。

しかし管理面から見ると、あくまで専用の装置であるX端末は、ワークステーションに比べて保守管理の手間がはるかに小さくなることが期待できる。例えば、ハードウェアが単純な分故障の発生率も小さいであろうし、障害時の復旧も容易である。また、ソフトウェアの保守も不要である。さらに、電源投入、切断、リセットなどの操作（あるいは誤操作）に神経を使わなくてすむという、利用者、管理者双方にとって大きな利点もある。

性能面に関しては次節で述べるような実験を行った結果、基本的なプログラミング演習程度の利用であれば、現在のワークステーションの平均的な性能で10台程度の端末を同時にサービスすることは可能であると判断した。したがって上で述べたような管理の容易さ、さらに導入コストの点から、X端末方式を採用することにした。

4.3 多人数使用時の応答性能の測定

システムの構成を決定するにあたっては、1台のワークステーションで何台の端末（何人の利用者）に対してサービスができるかを見積もる必要がある。しかしこれは、システムの利用形態によって大きく変動する可能性があるため、明確な指針となるデータがない。そこで我々は、実際の学生演習を想定した環境を設定し、その中でのワークステーションの応答性能を調べることにした。

実験システムとして、18台の既設のX端末（伊藤忠エレクトロニクス社 CIT-X）と1台のワークステーション（SPARCstation 2）、そしてファイルサーバ役のワークステーション（SPARCstation 1）を同一のサブネットに接続したものを使用した。

実験では、端末数（利用者数）を3、6、9、12と変えながら以下のような操作を全員一斉に行い、各利用者ごとに実行時間を測定した。

- (1) ログイン**
- (2) 約 120 KB のファイルに対する文字列操作

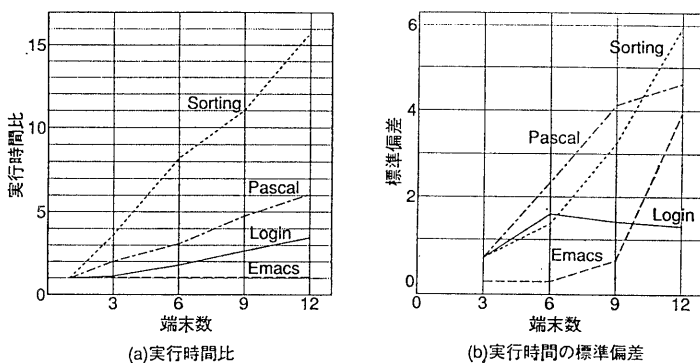


図2 端末数に対するシステムの応答性能

Fig. 2 Effect of the number of terminals on response time.

(tr, sort, uniq)

(3) 15 KB, 600 行弱のファイル (Pascal プログラム) を対象とする emacs エディタの各種操作*

(4) Pascal および C のプログラム (500 行強) のコンパイル

(5) ログアウト**

なお*印の測定項目は手で計時を行い、それ以外については csh の time コマンドを使用した。

測定結果のうち4項目について、端末数の増加に対する実行時間比（実行時間の平均値と端末数1の時の実行時間との比）の変化を図2(a)に、また端末数の増加に対する実行時間の標準偏差の変化を図2(b)に、それぞれ示す。なお端末数1のときの各実行時間は(1) 13秒、(2) 2.1秒、(3) 2.0秒、(4) 9.6秒、(5) 13秒と妥当な値であるため、これらを応答性比較の基準とした。

図2(a)より、各処理の実行時間は、端末数にほぼ比例して増加していることが分かる。実行速度の平均値で見ると、12端末までの範囲では、端末数にほぼ比例した反応を示しており、急速に性能が悪化するなどの現象は見られない。しかし図2(b)を見ると、処理によっては端末数が9から12に増加した時点で、実行時間の標準偏差の値が急に大きくなっている。これは各端末での実行時間のばらつきが大きくなり、一部の端末ではかなり反応が悪くなっていることを反映

* ログインウィンドウにパスワードを入力し終ってから、tvm, kterm 等いくつかのクライアントが起動し、kterm でシェルへの入力ができるようになるまでの時間。

** ログアウト処理を開始させてから、起動していたクライアントを終了し、次のログインウィンドウが出るまでの時間。

している。

こうした結果から、学生の一斉操作を想定しなければならぬ集合教育用としての条件のもとでは、1台のワークステーションでサービスできる端末数は9台程度であることが分かる。実際には本実験のように全部の端末がタイミングを合わせてコマンドを投入することはないので、入力タイミングがずれることによるシステム負荷の軽減を考慮して、12台程度が限度であると判断した。またこれとは別に、一斉に端末操作（ウィンドウの移動）を行ったところ、12台の時、一部の端末のウィンドウが追従できずにハングアップする現象が起り、ネットワークトラフィックの面でも限界に達することが判明した。そこで実際の設計指針としては、さらに若干の余裕を見込んで、10台程度を上限と考えることにした。

4.4 ファイルシステム構成

分散システムにおけるファイルシステムの配置方式としては、各ワークステーションと共に分散配置する方法と、ファイルサーバを置いてそこに集中させる方法とを検討した。

純粋に入出力性能だけから見れば、システムのディスク資源はすべて各ワークステーションに直接接続するのが最も良いはずである。しかしこの場合、ディスクがシステム全体に完全に分散して配置されることになるため、ファイルの整合性の確保やバックアップの方法など、管理面での問題が生じてくる。また、利用者ファイルも分散配置すると、すべての端末で同じ利用環境を提供するという条件を満たすためには、システム全体に渡って非常に複雑なファイル参照関係を設定しなければならないため、運用上も、また性能的にも問題がある。

逆にファイルサーバを置いてすべてのファイルをそこに集中させることにすれば、管理は一元化されて容易になるものの、システム稼働時に全ワークステーションからのファイルアクセスがファイルサーバに集中することになり、性能的に大きな不安がある。これでは分散システムの利点を失うことになりかねない。

そこで本システムでは両者を組み合わせ、ワークステーションのローカルディスク上とファイルサーバのリモートディスク上という、二段階のファイルシステム構成をとることとした（図3）。使用頻度の高いシステムファイルや作業領域などはローカルディスク上に置き、利用者ファイルや使用頻度の低いユーティリティなどのファイルはリモートディスク上に配置する。

このようにすることで、全体としてのファイルアクセス性能の低下を少なくすると共にファイルサーバへのアクセス集中の可能性を小さくし、ローカルディスクの容量も小さく抑えることができる。また利用者ファイルについては、1か所で集中管理できる利点も失わずにすむ。なお、この方式において、ローカルに配置するシステムファイルは複数存在することになるが、それらの更新は管理者による限定的なものであり、更新の頻度も少ないため、4.7.2項に述べる方法によって一元的に管理することが可能である。

4.5 ソフトウェア構成と利用者インターフェース

システムに導入するソフトウェアとしては、システムに付随するUNIXツール群やプログラミング演習に必要な各種プログラミング言語系、エディタ（Emacs）のほか、電子メール、ニュース等の通信ツール、TeXなどのドキュメント作成ツールも準備する。また、授業のための支援機能として、ログインの有無による出席管理、電子メールによるレポートの管理⁹⁾、および教官の指示ウィンドウ（文字と画像を含む）を学生全員が参照できる画面管理⁹⁾などのソフトウェアを開発し提供する。

そのほか、基本的にはウィンドウシステムによるGUIを提供するが、学生に極力基本的な「生」環境を提供するという立場から、標準ではXウィンドウシステムの基本部分のみを提示し、OPENLOOKやMotifなどのより上位のインターフェースはオプションとすることにした。

また同様の理由から、ウィンドウ上のメニューシステムもあえて必要最小限のものに抑え、そのかわり利

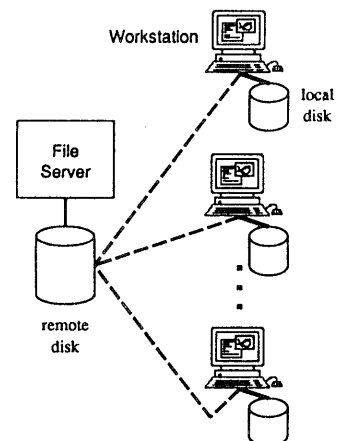


図3 ファイルシステム配置
Fig. 3 Allocation of file systems.

用者に対するカスタマイズや利用する上での制限は、極力なくすように配慮した。

4.6 ネットワーク構成

教育サブシステムには 200 台をこえる（飯塚地区）ワークステーション、X 端末が含まれる。演習などでそれらが一斉に動作すると、全体として非常に大きなトラフィックが発生する。それらが単一のネットワークに集中すれば、そのネットワークを飽和させてしまうことは十分予想できる。そうすると、端末とワークステーション間、そしてワークステーションとファイルサーバ間の通信が阻害され、システム全体としての性能が著しく低くするおそれがある。

こうした事態を避けるためには、

- ブリッジ、IP ルータなどの中継装置（ゲートウェイ）を導入し、ネットワークを適切な規模の複数のセグメントに分割する。
- トラフィックの発生要因を分析し、それらが各セグメントに均等に分散するようにする。

などの対策をとる必要がある。特にネットワークの分割は、ネットワークの一部に障害が発生した場合、その影響を局所化しシステムの他の部分に波及させないようにできるという点で、安全上の効果も大きい。

さて本教育システムの場合、トラフィックが最も大量に発生するのは、大人数での学生演習時であると予想できる。このとき主な要因となるのは、X 端末とそのホストとなるワークステーションとの間の通信、そしてワークステーション群とファイルサーバ間の通信（主にリモートファイルのアクセスによって発生するもの）の二つである。したがってこれらを分離すれば、ネットワークトラフィックの集中をかなり抑えることができる。

そこでまず、1 台のワークステーションと、それをホストとする複数台の X 端末をサブネットで結んだものを一つの構成単位（以下クラスタと呼ぶ）とする。そして各クラスタのワークステーションにはネットワークインターフェースを増設して、これをクラスタ内外間のゲートウェイ（IP ルータ）として用いることにより、図 4 のようなネットワーク構成を行うことにした。

通常の利用では、あるクラスタの X 端末がクラスタ外のホストと通信を行うことはない*。そのため、X

端末に関わるトラフィックは、ゲートウェアで遮られてクラスタの外に出て行かない。逆に、ワークステーションとファイルサーバの間などクラスタ外の通信は X 端末とは無関係であるため、そのトラフィックがクラスタの中に入ってくることもない。

この構成ではクラスタ内のトラフィックはほぼ X 端末に関わるものだけになり、その量は高々そのクラスタの端末数分である。これが性能上の許容範囲内であることは、4.3 節の測定時のトラフィックを観測することにより確認した。一方、クラスタ間接続セグメントのトラフィックは、リモートファイルアクセスを中心とするワークステーションとファイルサーバ間のものにほぼ絞られる。したがってこれに関わるのは、一つの教育サブシステムのすべてのワークステーション（およそ 20 台程度）のみである。これは一つのネットワークセグメントとして構成できる規模である。ただし、演習等で短時間に負荷が集中する場合については未知数の部分があるため、導入時において可能な限り、さらに複数のセグメントに分割することにした。

4.7 運用管理

4.7.1 端末とワークステーション

X 端末はワークステーションに比べて、管理にかかるコストを小さくすることができる。しかし、X 端末にもネットワークアドレスやホスト計算機の選択等いくつもの設定項目があり、数百台の端末についてそれらを個々に管理するのは容易なことではない。ワークステーションは端末に比べれば数で一桁少ないもの

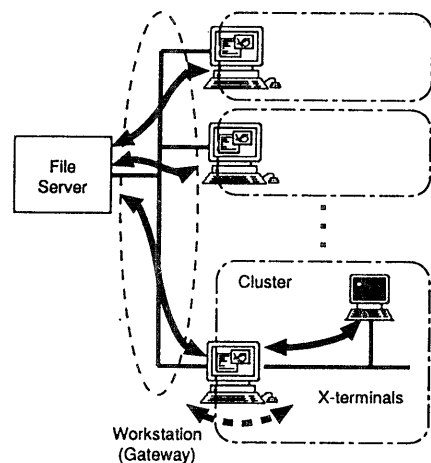


図 4 ネットワーク構成
Fig. 4 Network configuration.

* X 端末とクラスタ外のホストとの間の通信もちろん可能であるが、本システムの利用形態としては特殊な場合と見なすことができる。

の、その設定や日常の管理は端末よりはるかに複雑である。したがって、このシステムを適度なコストで運用していくためには、端末とワークステーションを極力一元的に取り扱えなければならない。

最新の X 端末では、各端末の設定をワークステーション上の設定ファイルの中に書いておくことができる。端末は起動時にそのファイルを読み込むことで、自動的に設定が完了する。本システムではこの機能を用いて、端末の設定はすべてワークステーション、さらにはそれらとネットワークで結んだ運用管理サブシステム上で管理する。設定の変更が必要な場合には管理システムで設定ファイルを変更し、その結果を各ワークステーションに配布すればよい。

また動作中の端末の制御や各種の動作状況のモニターも、ネットワークを通じて行うことができ、これらを利用すれば端末に関するほとんどの管理を、管理サブシステムを使って行うことができる。

さて、教育サブシステムを構成するワークステーションは利用者から見るとすべて同等に見えることを目標としている。そのため、システムの設定項目を整理してみると、個々のワークステーションごとに変更する必要があるのはネットワークアドレスやホスト名と、それらに依存する一部の設定ファイルのみであることが分かった。そこで、ワークステーションの設定（ローカルディスクの構成）は基本的にすべて同じものとし、個々に異なるパラメタだけをフロッピーディスク（シグネチャフロッピーと呼ぶ）上のファイルに入れて、起動時にその情報にしたがって、必要なシステムパラメタの変更を行うことにした。つまりシステムの導入が終了した時点では、すべてのワークステーションは同一の設定になっており、システムごとの差異は起動中に決定することになる。

この方式だと一度すべてのワークステーションのシグネチャフロッピーを作ってしまうと、その後のシステムの調整や変更は 1 台分だけ行えばよいので、ワークステーションの台数が多いことによる管理の負担はそれほど増加しない。

4.7.2 ファイルシステム

すべてのワークステーションで同一の環境を提供するために、各ワークステーションのローカルディスク上のファイルシステムは、すべてのワークステーションで常に同一の内容でなければならない（起動時カスタマイズ項目を除く）。そこでワークステーションの 1 台を標準機とし、そのディスク内容を他の機械に反

映させることにした。

システムの初期導入やバージョンアップのように、ディスク内容が大幅に変化する場合には、参照機のディスクの内容をすべてダンプして、他機のディスクにコピーする。しかし数個のファイルだけを追加、更新したい場合だと、この方法では、わずかな変更の割には手間がかかりすぎる。このような場合については標準機のファイルシステムを変更した後、それを UNIX のリモートファイル配布プログラム `rdist` を使用して他機に反映させる。この操作は管理者が手動で行う必要がある。しかし、ローカルディスクの内容をシステムファイルと基本的なライブラリ、ユーティリティのみとし、原則として新たなファイルを加えないようにすれば、この `rdist` 処理作業を行う負担は少なくすることができる。

システム運用開始後のフリーウェア、ユーティリティ等の導入は、原則としてファイルサーバ上の共有ディスク領域に対して行う。この領域であれば単一のファイル実体を導入するだけで、すべてのワークステーションから参照できるため、各ワークステーションのディスクに比べて導入、変更が容易である。

以上はシステム管理者だけが変更することのできるファイルシステムである。一般利用者は原則として、これらの場所にソフトウェアなどを置くことはできない。しかし授業や演習を行う際、教官が独自の教材として、ユーティリティやデータファイルなどを共用のファイル領域に置きたいということは頻繁に起こり得る。こうした要求に対して個々に対応することは、システムの整合性を失いやすく、また変更は慎重に行わなければならないため、非常に難しい。またフリーウェアなどの場合、依頼する側の方がそのソフトウェアについての知識や経験が豊富であるということも少なくない。

そこで我々は、授業担当者のユーザクラス（UNIX ではグループ）を設定し、共有ファイル域上に、そのグループに属する利用者が自由に書き込める領域（教材用領域）を設けた。そして、授業担当教官とその教官が認める補助者については、希望に応じて授業担当者グループ属性を持つアカウントを発行することにした。これにより授業関係者は、システム管理者を介さず自らの責任において、共用ファイル域にファイルを導入することができる。教材用領域の利用については、システムに問題を発生させるおそれがない限り管理者側から干渉されることはない。反面、安全性や利

用目的については、利用者の良識に期待する形になっている。

最後にファイルシステムのバックアップについて述べる。まずローカルディスクについては、一切バックアップは行わない。ファイルシステムの内容を共通化しているため、障害が発生した場合にも、他のワークステーションのディスクの内容をコピーすれば済むからである。

一方、利用者ファイルを保持するファイルサーバでは、約 20 GB のディスクを二つの系統に分け、一方を予備系として使用することにした。毎日（利用者のほとんどない）深夜に、正ファイルシステムの内容を対応する予備システムにバックアップする。このバックアップ作業はシステムが自動的に行うため、管理者はバックアップに関する作業を行う必要はない。ファイルシステムに障害が発生した場合には、当該ファイルシステムを予備系に切替えれば、最新のファイルは失われるものの、運用を続行することができる。

4.7.3 利用者管理

従来の本学のシステムでは、利用者識別名（UNIX の場合だと利用者名）は、その利用者の身分、所属を示す 2 桁のアルファベットと、学生番号の一部（学生の場合）または生年月日の一部（職員の場合）の 4 桁の数字とを組み合わせたものであった。しかし今日、利用者名は単にそのシステムにおける識別子というだけではない。広域ネットワークの中で利用者を特定す

るアドレスの一部であり、電子メールの利用などを通じて、広い範囲に公開されるものになっている。そうした点から従来の利用者名を見てみると、

- (1) 記号化された利用者名から本人を連想することがほとんど不可能。
- (2) 電子メール等でアドレスを指定する時に間違いやすい。また間違えた結果として、別の利用者を指定する可能性が高い。
- (3) 利用者側に自分の識別子を選択する余地がない。などの問題がある。(1)や(2)の対策として、利用者の姓名のローマ字表記をもとに、一定のルールに基づいて利用者を決定している例も多い。しかし利用者数が数千人という状況では、利用者名が重複しないようなルールを設定することは不可能に近い。そこで本システムでは利用者名の選択を、利用者自身に委ねることにした。

システムを利用しようとする人は、利用者登録の際に自分の希望する利用者名を申請する。すでに同じ利用者名での登録がないかどうかをチェックし、なければその申請を受け付ける。なお、この利用者登録の手続きは、オンラインでシステムと対話しながら行えるようにする。

ただし学部学生については、年度当初に多数の新入生の登録をする必要があること、システムに不慣れな状態で各人に申請作業をさせると、授業等に混乱を与えるおそれがあること、などから、従来通り管理側で

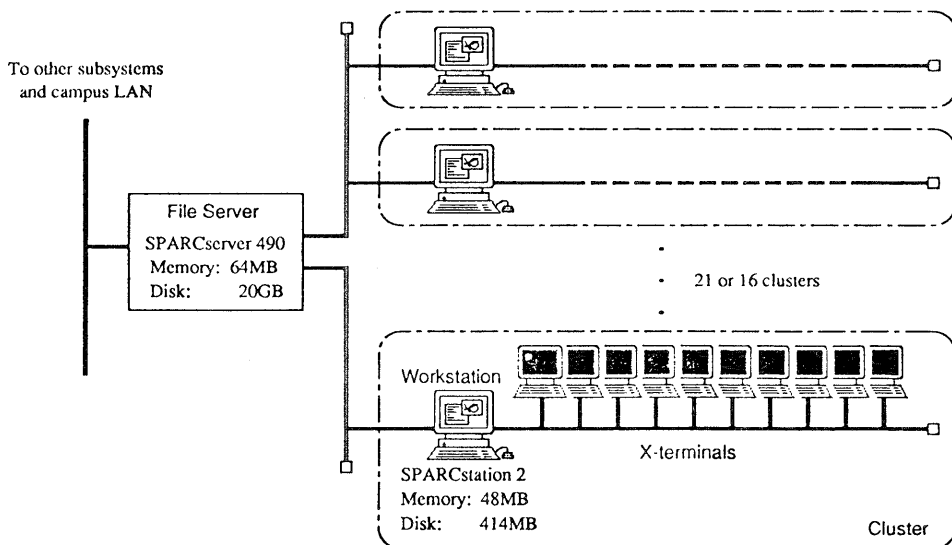


図 5 教育用サブシステムの構成

Fig. 5 Configuration of introduced educational subsystem.

規則的な利用者名を設定し、交付することにした。この場合も、利用者名の末尾2桁にその利用者のイニシャルを付加することにより、アドレスとして指定する場合に誤って他の利用者を指定してしまう可能性を小さくしている。

5. 実現結果と今後の課題

5.1 教育システムの構成

完成した教育システムの構成を図5に示す。4.3節の実験結果およびコスト上の要因などから、クラスタ(ワークステーション)あたりの端末数は原則として10台とした。飯塚地区のシステムは21クラスタ、戸畑地区は16クラスタで構成する。結局、教育システム全体でのワークステーションおよびX端末の台数は、

	WS	X端末
戸畑地区	16	157
飯塚地区	21	206

となっている。

また、導入したファイルサーバは三つのイーサネットインターフェースを内蔵できることから、クラスタとファイルサーバを結ぶネットワークセグメントは二つに分けた。この結果、セグメントあたりのワークステーション数は最大12台となった。ファイルサーバはまた学内ネットワークにも接続され、教育サブシステムと学内ネットワークとの間のゲートウェイとしての役割も果たす。

5.2 応答性能

このシステムを使って4.3節の応答性のテストを再度行った。導入前の測定時と比べてワークステーションの単体性能は変わっていないが、X端末は新しい機種に変わって表示性能が向上した。またファイルサーバもSPARCstation 1からSPARCserver 490になっている。ただし、今回の実験は約90名がシステムを使って演習をしている時間帯に行ったため、実際の運用状況を反映した結果にもなっている。

利用者数が10名(端末10台)の場合についての測定値(いずれも10名の平均)を、前回と今回とで比較

した結果を表1に示す。今回の測定での応答時間は、前回よりも若干早くなった項目と、数%程度遅くなった項目が半ばしている。X端末が新しい機種になったことは、このテストに関する限りシステムの応答性にはそれほど貢献していないようである。遅くなった項目に関しては、90人分の負荷によるファイルサーバの性能の低下が影響しているものと推測する。なお今回の測定でlogout処理時間が非常に長くなっている。現在のところ原因は不明であり、今後詳しく調査を行いたい。

このlogoutの項を除いては、現在のシステムの応答性能は、設計時の実験システムと同等の水準にある。

表1 応答時間の測定結果

Table 1 Measurement of command response times.

測定項目	平均実行時間(秒)	
	前回(導入前)	今回
login	31.7	37.3
ファイル, 文字列操作	48.5	43.6
emacs 起動	8.0	7.1
emacs カーソル移動	24.6	21.2
emacs セーブ	2.3	1.8
emacs 終了	1.6	0.2
Pascal コンパイル	50.3	54.7
C コンパイル	21.6	22.6
logout	16.7	33.0

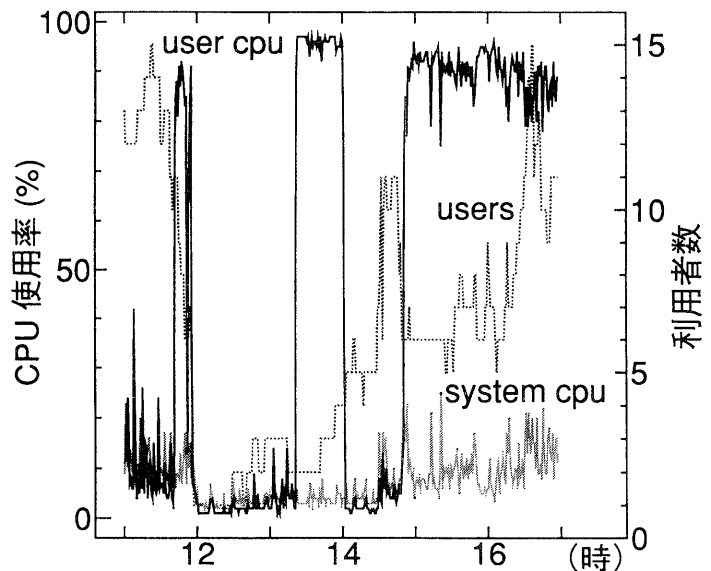


図6(a) 利用者数とCPU使用率(WS)

Fig. 6(a) Number of users and usage of the CPU time (WS).

り、全体としてはほぼ満足できる状態であるといえる。

5.3 最大稼働時におけるシステム負荷

運用を開始した教育システムは現在、90人のクラスが二つ同時に演習を行うという、設計時に想定した最大負荷に近い状態で使用されている。我々はこのような状態でシステムの各部にどの程度の負荷がかかっているか、いくつかの項目に関してシステム状態を観測することにした。なおこの観測時をふくめ、これまでのところ、授業に支障をきたすような性能上の問題は起きていない。

観測は飯塚地区の教育システムで、約90名のクラス二つが同時に演習する時間帯を中心に行った。すべてのワークステーションとファイルサーバで、主にUNIXのvmstat, netstatコマンドを使ってシステム状態をモニター（ネットワークは5分ごと、その他は1分ごとにサンプリング）した。

観測結果から、ワークステーションの1台とファイルサーバについて、

- 利用者数（ワークステーションのみ）
- CPU 使用率
- 主記憶残量
- 割り込み頻度（1秒あたりの割り込み回数）の平均
- ネットワーク入出力

についてのデータをグラフにしたものを図6に示す。なおデータ計測に際して、利用者数についてはshellウィンドウの数を数えているため、実際の人数よりも多くなっている。またネットワーク入出力に関しては、ワークステーションではクラスタ内部側（端末が接続している側）のインターフェース、ファイルサーバについてはすべてのインターフェース（ie0は学内ネットワーク、ie2, ie3は教育ワークステーション群と接続）のデータ数を計測した。

授業（演習）は12時50分から16時まで行われた。16時以降は自由利用で

あったが、受講生の多くが引続きシステムを使用していた。また12時から12時50分までは昼休みであった。

まずワークステーションの負荷（図6(a)(b)(c))を見てみると、授業時間の後半、本格的に演習が始まるころ（15時少し前）から、利用者プロセスのCPU使用率（user cpu）と主記憶の使用量が非常に増加

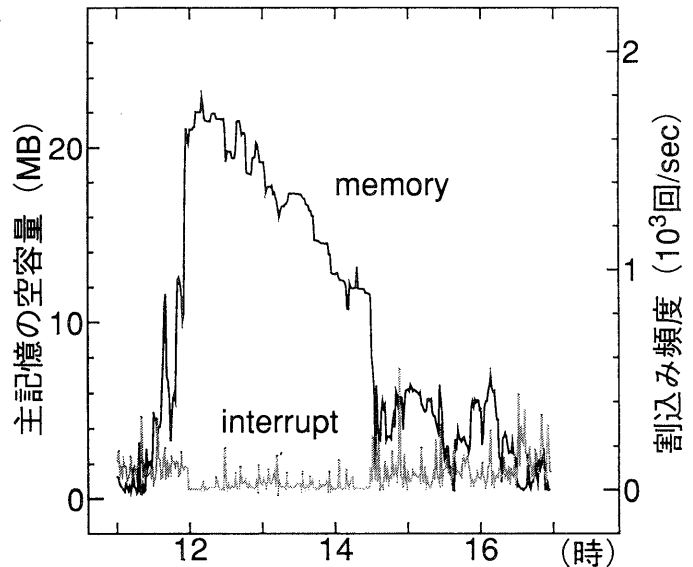


図 6(b) 主記憶と割り込み (WS)

Fig. 6(b) Remainder of the main memory and count of interruption (WS).

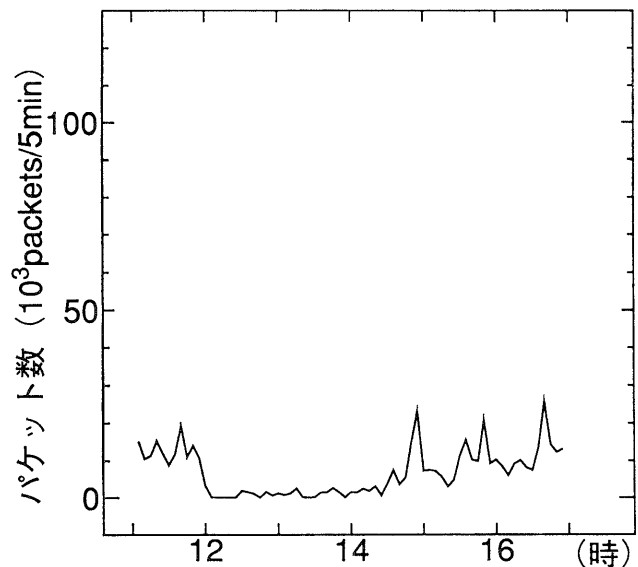


図 6(c) ネットワーク入出力 (WS)

Fig. 6(c) Network traffic at the network interface (WS).

し、両方ともほとんど使い切った状態になっている。同時期に、割り込みとネットワーク入出力の量もある程度増大しているが、それほど顕著ではない。

一方ファイルサーバの負荷 (図 6 (d)(e)(f)) ではどの項目についても明らかな時間的な変化は見えない。CPU 使用率におおむね 20% 前後の水準を維持している。一時的に 80% 程度に達することはあるものの、全体としては十分な余力があるようである。主記憶に関しても、観測時間内では最低でも 10 MB を残しており、こちらもまだ余裕がある。パケットの量も、17 時前後のピーク (おそらく、他のシステムとの間のファイル転送であろう) を除いては比較的低いレベルで安定している。割り込み頻度については多少高めに感じられるが、カーネル部分の CPU 使用率 (system cpu) が限界に達しているわけでもなく、問題ないものと判断できる。

結局、通常の利用形態 (演習等) での最大負荷時においても、ファイルサーバの能力、資源にはまだ余裕があることが分かった。心配していた、利用者ファイルアクセスの競合による過負荷状態も見られなかった。ワークステーションについても今のところまだ性能的な問題はなく、多人数での一斉演習にも対応できている。ただ CPU の能力や主記憶容量はほぼ完全に使われている時間帯もあり、将来的には不足気味になるかもしれない。

5.4 今後の課題

教育用システムの設計においては、システムにできるだけ冗長性を取り入れ、障害に強く復旧しやすいものとなるようにしてきた。しかしさまざまな制約により、

- (1) あるワークステーションに障害が起きると、従属する端末群がすべて使用できなくなる。
- (2) ファイルサーバに障害が発生すると、他のワークステーションや端末には全く問題がないにもかかわらず、システム全体が使

用不能になる。

という二つの点については有効な対策ができていない。

いずれの場合も、障害がファイルシステムのレベルであれば、ごく短時間で (授業に支障がない程度に) 復旧できる。ここで問題になるのは、より深刻なハードウェアの故障などの場合である。

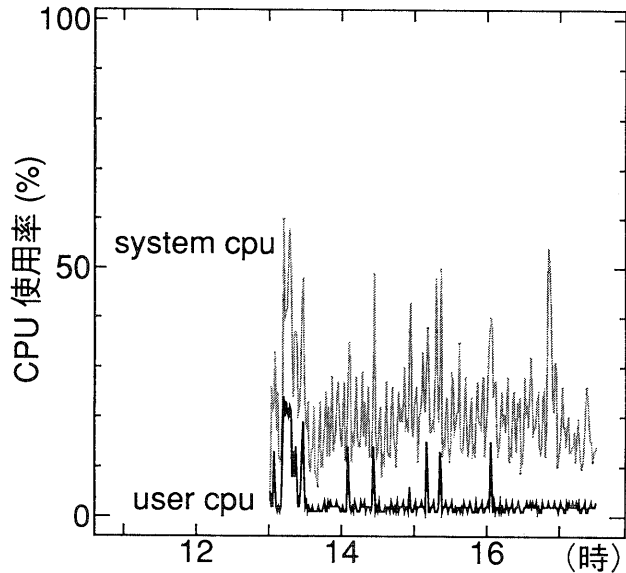


図 6(d) CPU 使用率 (サーバ)

Fig. 6(d) Number of users and usage of the CPU time (Fileserver).

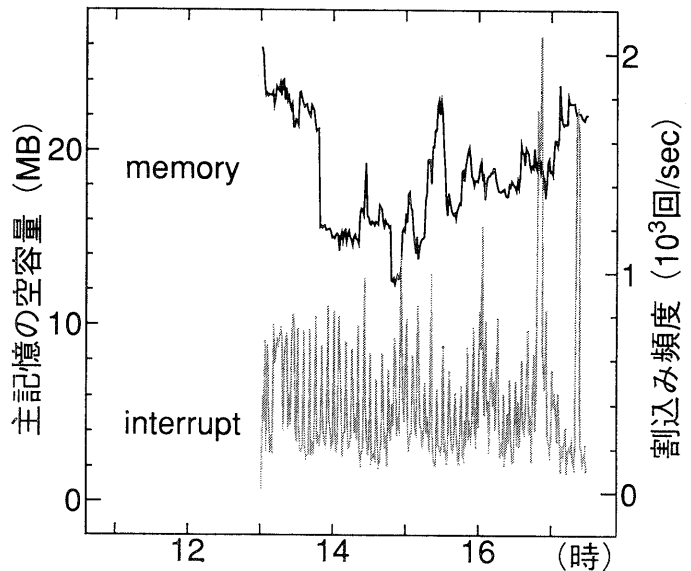


図 6(e) 主記憶と割り込み (サーバ)

Fig. 6(e) Remainder of the main memory and count of interruption (Fileserver).

(1)に対しては現在、ネットワーク的に適切なバイパス経路を設定し、端末を他のワークステーションに接続することを検討している。ただしバイパスへの切替え手順が複雑になること、また経路の設定いかんによっては端末ホスト間の通信がワークステーション間セグメントに流入して、ネットワークの性能低下の原因となるおそれがあること、などが問題である。別の案として、ワークステーションの予備機を待機させておくというものもあるが、こちらはコスト面の問題がある。

また(2)については、ファイルサーバを二重化するのが根本的な解決法であるが、コストの点で実現は難しい。しかしファイルサーバがない状態でも、ワークステーションとそれが持つユーティリティ群は使用できるため、各利用者にある程度の大きさの作業域を提供することができれば、授業や演習を続行できる可能性は大きい。現在この部分的なバックアップ方式の実現法を検討しているが、「作業領域」をどのようにして確保するか、システム復旧後、ファイルシステムをどのようにして通常の状態に戻すかなど、課題は多い。

ところで、定常的なシステムの監視、制御を支援する環境はまだ整備の途上であり、不十分な点も多い。今後はさらに統合的な管理システムの構築をめざす。また近年、ネットワーク管理を支援するツールやユーティリティなども発表されてきており、こうした製品の利用も積極的に検討していきたい。

6. おわりに

これまで見てきたように、本教育システムの機能と性能は、メインフレームを用いた従来の集合教育システムに代わるものとしては十分な水準にある。これまでのところ授業等での利用において特に問題となる点は見つかっていない。また授業以外に、自主学習のための利用も増加した。なかでも、市販の解説書等を使って、システムコールの応用やウィンドウベースのプログラミングなど、より高度な課題に取り組む者が多く見られるようになってきている。これは過度に手を加えない、標準的な利用環境を提供するという設計方針の効果が現れているものと考えている。

一方インフラストラクチャとしての計算機環境とい

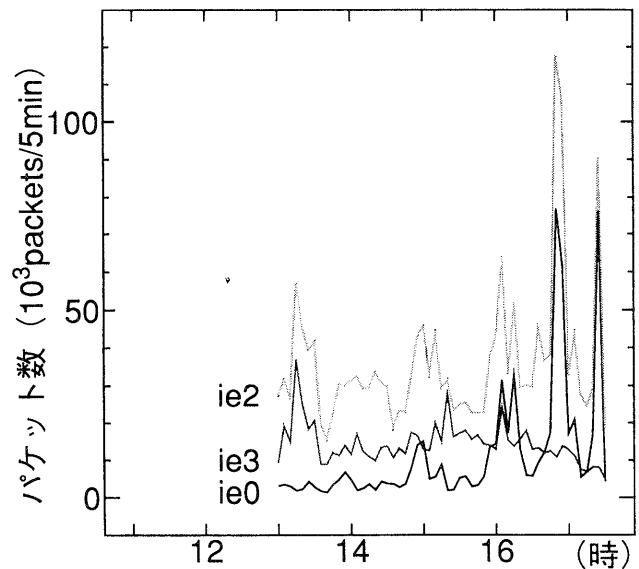


図 6(f) ネットワーク入出力 (サーバ)

Fig. 6(f) Network traffic at the network interfaces (Fileserver).

図 6 最大稼働時のシステム各部の負荷の変化

Fig. 6 Variation in system loads.

う面では、電子メール、ネットニュースといった、計算機を介したコミュニケーション手段の利用や、ネットワークを通じての他の学科システムとの相互利用などは活発化しており、基本的な第一歩は踏み出していると考ええる。

しかしこれまでのシステムの立ち上げ段階では、もっぱらシステムの安定性や基本性能の確保に力を注いできたため、分散環境や GUI を活用したより高度な教育支援環境や、インフラストラクチャとしての環境に必要な種々のユーティリティ、データベース等の整備はまだ始まったばかりである。

今回の大幅なシステムアーキテクチャの更新により、将来に向けての集合教育システムの土台はほぼ完成したと考えている。今後はその上に、実際に教育を担当する教員さらに一般の利用者の要求を反映して、より高度な環境を積み上げていきたい。

謝辞 センターシステムの仕様の検討にあたって多くの御意見、御助言をいただいた、九州工業大学システム/ネットワーク運用管理者グループ (kitnet) の皆様に心から感謝いたします。

参考文献

- 1) 安村通晃, 有澤 誠, 斎藤信男: コンピュータ

- リテラシー教育の一事例, 情報処理, Vol. 32, No. 12, pp. 1310-1317 (1991).
- 2) 村岡洋一, 笈 捷彦, 永田守男, 村井 純(編): 知のキャンパス—大学における情報教育環境—, p. 229, 共立出版, 東京 (1991).
- 3) 浮貝雅裕, 菅原研次, 三井田惇郎: 情報系学科新入生に対する導入教育とそのための演習教育環境, 情報処理学会論文誌, Vol. 33, No. 1, pp. 1-9 (1992).
- 4) 荻原剛志, 山口 英, 西尾章次郎: 新時代を迎えた情報処理教育環境, bit, Vol. 24, No. 4, pp. 376-385 (1992).
- 5) 大岩 元: 一般情報教育, 情報処理, Vol. 32, No. 11, pp. 1184-1188 (1991).
- 6) 大西淑雅, 中山 仁, 山之上卓, 藤木健士, 末永 正: 大学における大規模分散システムの構築(1)—利用面を重視して—, 計算機科学研究報告, 九州大学大型計算機センター, Vol. 9, pp. 15-24 (1992).
- 7) 中山 仁, 中村順一, 末永 正: 大学における大規模分散システムの構築(2)—運用面を重視して—, 計算機科学研究報告, 九州大学大型計算機センター, Vol. 9, pp. 25-34 (1992).
- 8) 山之上卓: シェルプログラミング入門応用例—授業レポート自動受け取りシステムについて—, 九州工業大学・情報科学センター広報, 第6号, pp. 57-68 (1993).
- 9) Yamanoue, T., Shimizu, M. and Fujiki, T.: Development of an Electronic Chalkboard for a Large Classroom by Parallel Programming and Its Application to English Classes, *Proc. of APITITE '94*, pp. 651-656 (1994).

(平成5年9月2日受付)

(平成6年9月6日採録)



中山 仁 (正会員)

1961年生. 1986年九州大学工学部情報工学科卒業. 1988年同大学院修士課程修了. 同年九州工業大学情報科学センター助手. 分散処理システム, 関数型言語システムの研究に従事. ソフトウェア学会会員.



大西 淑雅 (正会員)

1966年生. 1989年九州工業大学工学部情報工学科卒業. 同年九州工業大学情報科学センター助手. 分散システム, 計算機ネットワークの研究に従事.



末永 正 (正会員)

昭和24年生. 昭和47年九州大学工学部電子工学科卒業. 九州大学大型計算機センター, 同情報処理教育センターを経て, 平成元年九州工業大学情報科学センター助教授, 平成3年近畿大学九州工学部教授, 現在に至る. 計算機システムの性能評価に関する研究に従事. 九州大学工学博士. 電子情報通信学会会員.



有田五次郎 (正会員)

1939年生. 1963年九州大学工学部電子工学科卒業. 1965年同大学院修士課程修了. 同年九州大学講師(中央計数施設所属). 1984年九州工業大学工学部教授を経て, 1988年同情報工学部教授(知能情報工学科). 工学博士. 計算機アーキテクチャ, 並列処理システム, 計算機ネットワークの研究に従事. 電子情報通信学会, ソフトウェア学会会員.