

授業支援システム Ocean の機能と実現法

†齊藤明紀, †安留誠吾, †馬場健一, †重弘裕二, †西田 知博, †中西通雄,
‡原田 章, ††山井成良, ††松浦敏雄

†大阪大学情報処理教育センター

‡大阪大学人間科学部

††大阪大学計算機センター

††大阪府立大学学術情報総合センター

概要 大阪大学情報処理教育センターでは、1996年の機種更新を期に情報処理教育の授業を支援するためのシステム OCEAN を自らの手で構築しつつある。OCEAN は中核となる情報管理サーバと、それと連係する利用者登録、利用記録収集等のサブシステム、およびユーザーインターフェースを司るクライアントアプリケーションからなる。情報管理サーバは TCP/IP を用いてクライアントと通信を行なうが、単一プロセスで複数のクライアントにサービスするようつくられている。本センターでの利用者計算機の OS は NEXTSTEP3.3J であるが、支援システムは NEXTSTEP には依存しない形で作られている。また、NEXTSTEP や X などの GUI だけでなく無手順端末から支援機能を利用するクライアント・アプリケーションを作成することを考慮して設計されている。

The OCEAN Computer classroom management system: its function and implementation

†Akinori Saitoh, †Seigo Yasutome, †Ken'ichi Baba, †Yuuji Shigehiro, †Tomohiro Nishida, †Michio Nakanishi, †Akira Harada, ††Nariyoshi Yamai, ††Toshio Matsuura

†Education Center for Information Processing, Osaka University

‡Faculty of Human Science, Osaka University

††Computation Center, Osaka University

††Media Center, Osaka City University

Abstract Osaka university's Education Center for Information Processing is now constructing a classroom management system. The system consists of central information server and a few cooperative subsystem such as user registration system and attendance information gathering system. The central information server is a single process communicating with multiple client applications with TCP/IP. Our client computer are running NEXTSTEP3.3J, but this management system does not depend on NEXTSTEP. It is designed to allow not only NEXTSTEP clients but also X or character terminal based clients.

1 はじめに

本報告では、大阪大学情報処理教育センターで現在作成中の授業支援システム Ocean について述べる。本システムは、学生への教材や課題などの伝達、電子的なレポート受け付け、出席の自動管理などを行なうものである。

本センターの登録利用者数は 1 万 3000 人以上、端末数は 562 であり、本センターで行なわれる講義や演習は年間で延べ 70~80 である。授業支援システムはこのような規模に対応するものでなければな

らない。

Ocean は 授業支援用データベースサーバーである MIX と、それを保管するいくつかのサブシステム、および、利用者用計算機で動作するユーザーインターフェースアプリケーションからなる。

2 設計方針

本センターでは NEXTSTEP3.3J を採用しているが、それは学生の使用するクライアント計算機で

あり、管理用計算機やサーバー計算機は他の機種の UNIX ワークステーションである。そこで、ユーザーインターフェースアプリケーションを除いては、NEXTSTEP に依存しない形態で実現することを目標とした。さらに、本センターに独自の仕様はできる限り排除し、やむを得ない部分はサブシステムとしてくり出すようにしている。また、センター職員や教官のために、NEXTSTEP 以外のウィンドウシステムでのクライアントプログラムにも対応できることが望まれる。

クライアントは、TCP/IP が利用出来るマシン全てで利用出来るシステムにしたい。ただし、何らかの形で利用者認証が可能なものに限る。

システム構成としては、可能な範囲で、依存性の少ないサブシステムの集合体とする。これは、以下の理由による。

- 単一の巨大システムは開発、保守が困難。
- サブシステムの集合体であれば、出来上がった部分から順に運用を開始できる。

2.1 構造設計

UNIX ではユーザのグループに対してファイルのアクセス制御を行なうことができるが、一人の利用者が所属できるグループの数は 16 程度である。一方、講義一つに対して

- 教官
- 教官とティーチングアシスタント
- 教官とティーチングアシスタントと受講生

という 3 つのアクセス権グループは少なくとも必要になると考えられる。よって、UNIX のファイル保護機能そのままでは不足である。

そこで、情報サーバ (MIX サーバー) をサーバー計算機上で動作させ、利用者の起動するクライアントプログラムとの通信で情報サービスを行なうという形式にした。ただし、効率の観点からクライアントが読み出す情報のうちアクセス制御の不要なものは、NFS を用いることにする。

MIX サーバーは、基本的には以下の機能を持っては良い。

- 利用者に情報を提供する
検索により、情報のうち必要な一部を提供する

- 利用者から情報を受けとって保管する
- 保持する情報の一覧を利用者に提供する

これに付随して、次のような機能も必要である。

- 各情報項目の読み出し、書き込みに関して、操作可能なユーザ群を設定できること。
- 動作記録を残すこと
- データの保守管理用ツールを備えること

2.2 データベース設計

現在のシステムでは、アプリケーションプログラマに見せる情報項目の名前は、その情報が格納されたファイルのパス名と 1 対一に対応している。以下に、そのうちの重要なものをあげる。

- `/global/classes/`: 講義一覧、教官からの講義逆引きデータなどが置かれるディレクトリ。
- `/global/rooms/`: 教室一覧と、それぞれに置かれたホストリストが置かれるディレクトリ。
- `/accounts/printers/`: プリンタの利用状況が格納される
- `/accounts/users/yyyymmdd/tt/room`: yyyy 年 mm 月 dd 日の tt 時限の第 room 番教室での利用者毎の総ログイン時間が格納される。
- `/classes/classID/`: 学生リスト、開講日/休講日情報、学生への課題、教材、連絡事項、学生から出されたレポートといった、各講義に対応したデータが格納されるディレクトリ。
- `/classes/revclasses`: 学生のログイン名をキーとして、その学生が受講している講義 ID リストを取り出すためのデータベース。

3 MIX サーバーの実現

MIX サーバーは特定のディレクトリ以下に格納されたデータへのアクセスを提供する。また、学生リストなど高速な検索が必要なものに関しては、BTREE 形式のファイルとするが、基本的にはプレーンテキストファイルを使用する。

3.1 データ形式

データはテキスト形式で表現する例えば授業 ID などの数も、数字列で格納する。これによる性能低下はごくわずかである。一方、UNIX にはテキストを処理するツール群やライブラリ関数が多く備わっており、テキスト形式にすればそれらを用いることができる。また、データの手動修正をテキストエディタで行なうこともできる。

文字コードは、日本語 EUC を用いることにした。データ中の改行コードは UNIX に合わせて 0x0a(LF) とした。

多くのプラットフォームに対応するグラフィックデータの形式の選定は難しい問題である。現在想定した範囲では、授業支援システムは一部の例外を除き、テキストデータしか扱わない。例外は、教官が学生に提供する教材、および学生が提出するレポートである。そこで、本サーバはそれらは単なるバイナリファイルとして取り扱い、内容の解釈や変換は行なわないこととした。

セキュリティに関しては、以下の配慮を行なった。

- シンボリックリンクを作る機能は提供しない
- パス名中に . や .. が含まれたものは不正な情報項目名として拒絶する。

3.2 実装形態

通常の UNIX のサーバは、クライアント毎に個別のプロセスが起動してサービスを行なう。しかし、本システムでは多くのクライアントが一斉にアクセスする可能性があり、サーバへの負荷が心配である。また、独立したサーバプロセス感でのデータファイルへのアクセス競合の調停メカニズムを実装するのも難しい。

そこで、本サーバは 1 プロセスで複数のクライアントのサービスを行なう形態をとることとした

3.3 アクセス制御

アクセス制御は、ディレクトリ毎にアクセス権定義ファイルを置いて行なう。アクセス権制御ファイルは、グループ定義ファイルを参照する。たとえば、番号 101 の講義に対して、教官、教官+TA、教官+TA+受講生という 3 つのグループを定義するには以下のように記述する。

```
c101m:1:sp-001wm
```

```
c101t:2:+@c1001m,+classes/1001/stafflist
```

```
c101s:3:+@c1001t,+classes/1001/studentlist.bt
```

コロンで区切られた第 2 フィールドはグループ ID であるが、UNIX のグループ ID とは関係内。

3.4 利用者認証

項目毎のアクセス制御のためには、利用者認証の仕組みが必要である。

現在公開鍵暗号など高度な手法が一般的になってきつつあるが、暗号化なしの NIS や NFS 等を運用しているセンター内部で、授業支援システムだけ高度な認証機構を持たせても意味がない。

利用者認証は、ポート番号、IP アドレスによるホスト認証、ident プロトコル [1] による利用者認証、パスワード、チケットを組み合わせたものを使っている。

3.4.1 TCP によるアクセスの認証

クライアントの IP アドレスを調べ、信用できるホスト (センター内のホスト) であれば、そのその TCP コネクションに関して ident プロトコルで利用者を名を得る。センター外のホストに関しては、当センターとはログイン名が独立に管理されているだけでなく、ident による回答が必ずしも信用できない。この場合は、パスワードによって認証する。TCP の場合は、セッション毎に 1 回だけ認証を行なえばよい。

3.4.2 UDP によるアクセスの認証

UDP によるアクセスの認証は、サーバが発行するアクセスチケットを用いる。アクセスチケットの発行は、一旦 TCP で接続することで前項の TCP 認証を行ない、チケットを発行する。

3.5 通信ライブラリ設計

通信ライブラリは、汎用の情報読み出しアクセス関数は通常のファイルアクセス関数に似せた仕様で作成した。これはプログラマへの利便と、開発の効率化に役立つ。

例えば、mix_open() は open(2) と同じくファイルディスクリプタを返す。以降は、通常のファイル操作関数を用いてアクセスできる。

サーバーが完成する前のクライアントのデバッグは、サーバーと通信するのではなくローカルファイルに対する操作関数を呼び出すダミーライブラリを用いて簡単に行なえる。

4 クライアント・アプリケーション

4.1 教官ツール

図 teacher-scr は、教官用 GUI ツール(NEXTSTEP)である。講義選択ウィンドウ(左上奥)で講義を選択するとメインウィンドウ(右)が現れる。メインウィンドウで[受講性登録/修正]ボタンを押し、学生登録ウィンドウを表示させた場面である。学部学科と入学年度をマウスで選択すると、そのクラスの学生一覧が現れる。ウィンドウの下半分には現在受講登録されている学生の一覧が表示され、マウス操作で学生の登録/削除が行なえる。

学生の出席は、講義時間(大阪大学では90分)の50%以上ログインしていたかどうかによって行なっている。判定基準は、教官毎に変更することができる。

4.2 学生ツール

図 student-scr は、学生用 GUI ツールで、レポートを提出しようとしているところである。あらかじめ、ファイルに作成したレポートをセーブしておき、そのパス名を選ぶ。複数のファイルを提出したり、あるフォルダ内をすべて提出することもできる。

4.3 お知らせツール

オンラインの学生への通知システムは、通常は学生が能動的に読む操作をせねばならない。

当センターではログインウィンドウに改造を施し、ログインしていない状態でもセンターからの重要なお知らせは画面に表示されるようにしている。

5 出席情報収集システム

授業支援システムでは学生の出席管理¹のために、ログイン時間を収集する必要がある。ただし、講義開始以前からログインしている学生、講義終了後ログアウトせずに自習利用を続ける学生、何らかのト

¹本システムで提供する出席管理機能を利用するかどうかは講義毎に担当教官の判断による

ラブルで席が変わってログインし直す学生に対応せねばならない。また、学期の当初の講義では講義中にログイン、ログアウトを数回繰り返すこともあり得る。

出席情報収集システムは、ログイン情報収集システムと出席情報への変換システムからなる。

5.1 ログイン情報収集システム

UNIX のログイン記録は /usr/adm/wtmp というファイルに記録される。各クライアントホストでは開館時間中は1時間毎に /usr/adm/wtmp を調べ、増加分をサーバーに転送する。この際、クライアント側の責任でファイル形式を big endian に変換する。これは、当センターのクライアントホストがバイトオーダーが異なる機種が混在しているためである。

サーバーの負荷を平均化するため、転送を行なう時刻はホスト名と IP アドレスを元に簡単な算術操作を行ない、分散するようにしてある。

何らかの理由で転送が失敗した時には、次回の転送で最後に成功した転送以降に増えた分が転送される。

5.2 出席情報変換システム

本システムでは該当時限の合計ログイン時間をもとに出席判定を行なうことにした。

1日1回深夜に、集まった wtmp から console に関するレコードだけを取りだし、各時限の開始/終了時刻で分割する。さらに、重複ログインによる二十計数を避けながら教室毎に集計する。

MIX サーバーに登録するのは、ログインしていた合計時間(分)である。出席欠席の判定はユーザーインターフェースプログラムが行なう。

6 ユーザー登録システム

大阪大学の全学生が利用権を持つ当センターでは、ユーザー登録の省力化が重要な課題である。

MIX サーバーが保持する学生の情報は、氏名(漢字、かな、ローマ字)、入学年度、出席番号所属学部学科、である。

例えば、同姓同名の学生を識別して受講性登録することを考えると、以上の情報は最低限必要なも

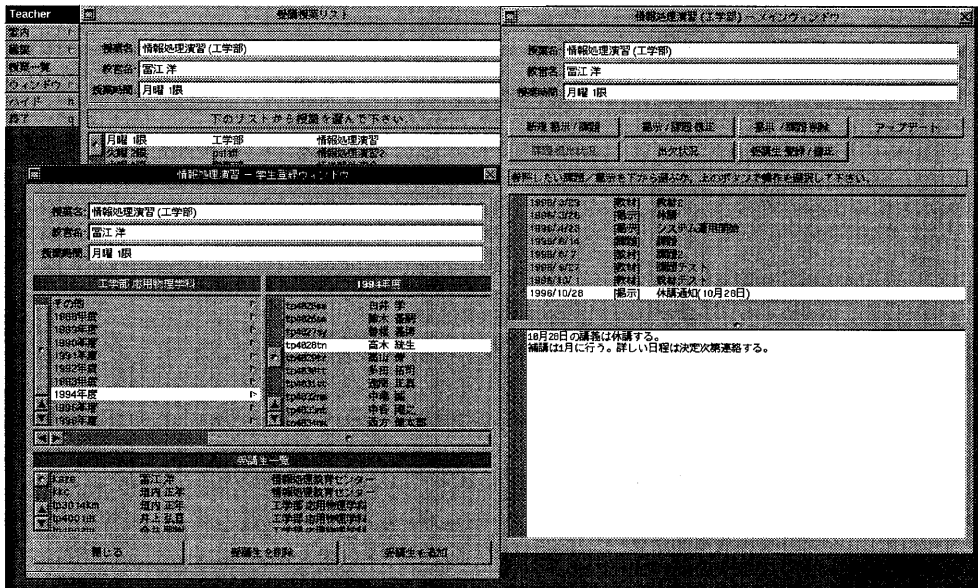


図 1: 教員用ツールの画面

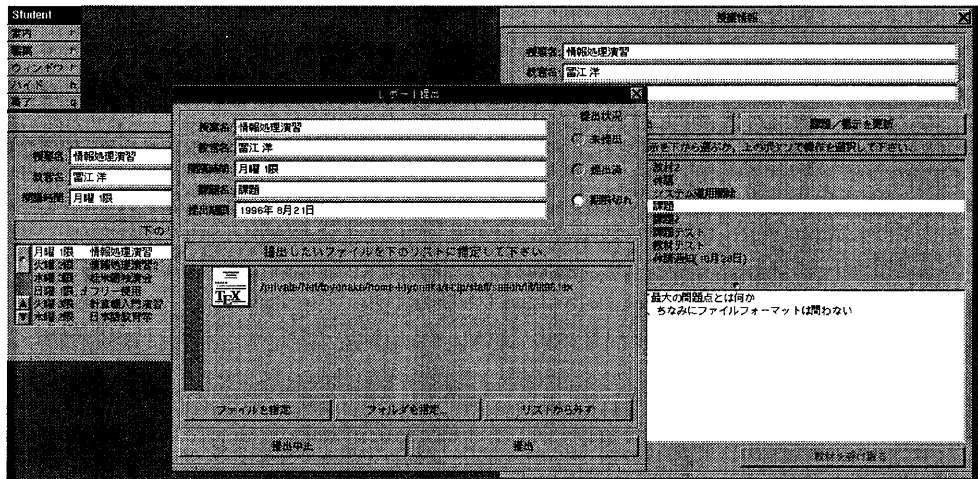


図 2: 学生用ツールの画面

のである。学生のプライバシー保護の観点からも、必要のない個人情報は扱わない方が安全である。

処理効率のため、所属学部学科はコード化する。学部学科コードは、募集定員の枠に対応している。募集が学部単位であれば学部、学科単位であれば学科をコード化している。このコードは大阪大学の内部で用いている3桁の学部学科コードをもとに一部改変を加えたものである。

6.1 ログイン名

登録を自動化するには、ログイン名などは自動的に生成できる採用せねばならない。

旧システムでは共用部での学生番号である8桁の数をそのままログイン名としていた。しかし、数字だけからなること、および入学年度を元号の下ひと桁で表していることがいくつかの不都合の原因と

なっていたため、英字で始まる8文字のログイン名に変更した。その内容は先頭から順に以下の通りである。

英字2桁：学部学科に対応する。

英数記号1桁：w,x,y,z,0,⋯,9はそれぞれ1996、1997、⋯、1999年度入学の学部学生を表す。vはその他の学生(大学院生、研究生)、-(マイナス)は職員を意味する。

数字3桁：出席番号を表す。教官や研究生、大学院生など「出席番号」の概念が当てはまらない利用者に関しては、登録を受け付けた順に1からの連番を割り振る。

英字2桁：チェックコード。学生の名前のローマ字表記から2字を取っている。

6.2 ユーザーID(UID)

NEXTSTEPを含め、多くのUNIX系のOSのユーザーIDは16ビット整数であるが、本センターの規模からして、学生の学部学科、入学年度等をコード化してUIDとするような関係性は作成できない。そこで、ある範囲のUIDを順に割り振ることにした。現在は、範囲は5000から24999である。抹消した利用者のUIDがすぐに再利用されると、誤抹消を行なった際の復旧作業が複雑になる。そこで、割り当てはラウンドロビン方式としている。

UID割当システムは単純なプログラムで、最後に割り当てたUIDをあるファイルに記録している。要求があると現在のユーザー登録状況を調べ、使われていないUIDを前回の続きの番号から順に出力する。

6.3 新入生登録システム

本センターでは、4字のランダムな文字列を印刷した紙を封筒に入れて学生に配っている。これはパスワードの一部で、学生の誕生日(月と日)の4桁の数字を連結したものがパスワードとなる。こうすることで、遺失した封筒を拾ったものが内部に書かれたパスワードを用いてログインすることはできない。また、クラス名簿などから誕生日を知ってもパスワードを知ることはできない。

新入生登録システムは、大学本部より提供されたテープを読みとり、EUCのテキストファイルに変

換する。学部学科コードとログイン名の先頭2文字の変換表と、名前の読み(JIS X02010)をローマ字変換することで、MIXサーバーに登録するデータは作成できる。

これからさらに、UNIXの利用者登録データを生成する。まず、UID生成システムから人数分のUIDの割当を受け、個々のユーザに割り付ける。続いて、封筒の印刷時に用いた4桁の文字と学生の誕生日データからパスワードを生成し、crypt関数で暗号化して/etc/passwdファイル形式のユーザー登録データとする。

大学院生や留学生や研究生などは誕生日データは得られない。そこで、パスワードはランダムな8字とし、一人づつ事務室の窓口で学生証を確認し、ログイン名とパスワードの印刷された利用者登録通知書を手渡している。

パスワードを忘れた学生に対しても、同様にセンターでランダムなパスワードを再設定している。

本センターでは利用者管理にはNISを用いており、24時間いつでも利用者がパスワードを変更する可能性がある。そこで、利用者によるパスワード変更と利用者登録システムによるパスワードファイルの更新が競合しないような機能をNIS(rpc.yppasswdd)に追加している。また、講義でパスワードの変更を教える際など数10人~100数10人が短時間の間にパスワード変更操作を行なうことがある。rpc.yppasswddにはこのような高負荷にも耐えるような改造を施したものをを用いている。

7 まとめ

当センターで現在開発中の授業管理システムOceanについて、設計および実装形態を報告した。Oceanシステムは完成したサブシステムから逐次実用に供されている。

Oceanの中核となる情報管理機能を司るMIXサーバーは、任意のユーザーグループを定義してアクセス権を設定する機能を持ち、TCPとUDPを併用することで高負荷に耐える設計となっている。

また、MIXサーバと連携する利用者登録システムや出席記録収集システムも作成した。

参考文献

- [1] M. St. Johns: "Identification Protocol", RFC1413, 1993.