



計算機リテラシー

— 知能情報工学科における実施経験 —

中村 順一¹

1 はじめに

情報工学部の設立から早 7 年近くが立ち、平成 6 年 3 月には 2 期生が修士を修了した。設立時に作成されたカリキュラムは、もちろん、色々と考えたすえ、決定したものではあったが、実際に実施した経験から、問題点もいくつか出てきた。教えるべきことが欠けている、教える時期が早すぎる、など内容面の問題もあったが、“教官側は簡単に分かると思っていた計算機の操作方法が学生にとっては案外難しかった”という事実もあった。更に、ここ数年の計算機環境の急速な変化により、操作の点で、従来は難しかったこと（コマンド等）が簡単になり、逆に従来は考えられなかった複雑な操作（対話的 GUI 操作）が必要になったという点がある。

平成 5 年度から新しいカリキュラムで教育を行うことにする際、検討の結果、プログラミングだけではなく、“操作方法”を教える授業が必要であるということになった。そこで「計算機リテラシー (literacy, 読み書きの能力)」という、もっぱら、計算機の操作方法を教える授業をカリキュラムに組み込んだ。知能情報工学科の情報科目は、図 1 に示すようにプログラミング系と計算機システム系の 2 つの流れから構成されているが、計算機リテラシーは、プログラミング系の最初の科目である²。

本稿では、計算機の“操作方法”だけを教えるという従来はなかった授業を実施した経験を“日誌風”に紹介する。

2 講義の必要性

着任して最初の卒研究生が配属された際、教官の多くは大きな期待を持っていたはずである。筆者の出身の講座では、卒業研究の最初はまず、講座にある計算機の使用方法を教えることから始まった。これに対して、知能情報工学科では、1 年生から情報科学や人工知能関連の科目を教えており、更にワークステーションを使って実験演習を行っているので、すぐにでも研究そのものが開始できると考えていた。しかし、現実には、login はできるものの、エディタでごく簡単なプログラムを書くのがやっとの学生が少なからずいた。

¹情報工学部 知能情報工学教室, nakamura@ai.kyutech.ac.jp

²科目は、自然科学科目、情報科目、対象分野科目に分かれている。言語理論や情報理論などは対象分野の科目として教育している。なお、計算機リテラシーはカリキュラムの分類上は対象分野科目である。

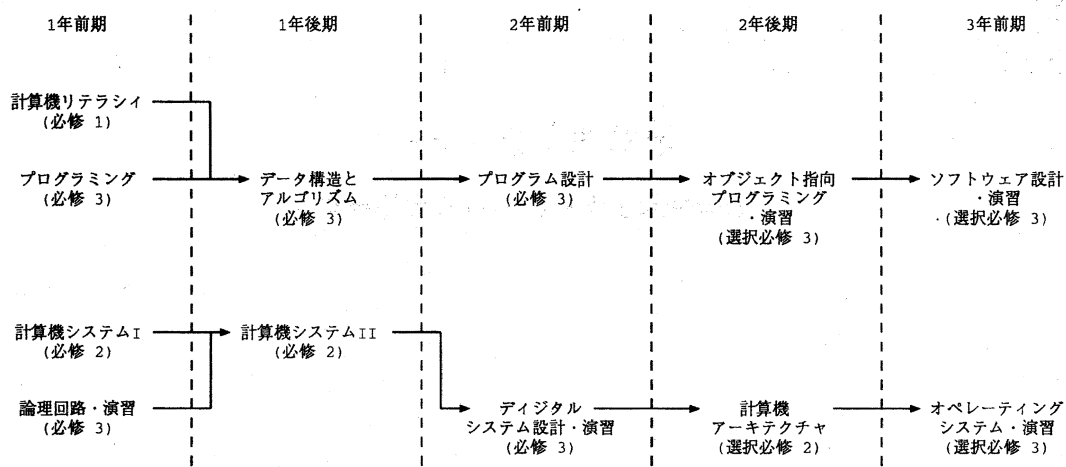


図 1: 知能情報工学科科目系統図 (一部)

しかし、学生の身になって考えると、本人の努力が足りないとは言いきれないことがある。“操作方法”に習熟するためには、それなりの時間、集中して演習を行わなければならない。しかし、以前のカリキュラムでは、“操作方法”は、プログラミングの“片手間”としてしか教えることができなかった。loginができて、それなりにエディタやコンパイラの使い方が分かれば、曲りなりにもプログラムの演習ができる。授業・演習で教えたいのは“操作方法”ではなく、プログラミングそのものである。最低限の操作ができれば、それで良しとなる。しかし、演習内容が高度になるにつれ、操作に習熟していなければついていけない状態になり、“落ちこぼれ”，そのため更に計算機を利用する時間が減り、操作が習熟できない、という悪循環が発生しているのではないか。

また、計算機上で動作する具体的なアプリケーションをあまり知らない状態で、リストだソートだといった講義を受ける。プログラムを作成していることが現実とは何の関係もないように見え、興味をなくす学生がいるのも無理はない。

更に、情報科学センターの計算機は、授業がなければ自由につかえると言っても、平均的な学生にとっては、他の授業を受けるのに忙しい。夜であれば自由に使用できると言っても、レポート作成のような具体的な課題でもあれば別であるが、夜まで“操作方法”を自発的に学習することはなかなか期待できない。

このことから、

1. まずは道具の使い方を覚える、
2. ソフトウェアに対する具体的なイメージを与える、
3. できるだけ計算機を操作する時間を増やす、

という必要を感じ、もっぱら“操作方法を教える”授業を開講した。

3 シラバス

科名: 計算機リテラシー (1 年前期必修)

担当者: 中村順一

講義内容: 計算機システムの基本的な利用方法を学習する。これには、オペレーティングシステムの利用方法、プログラム・文書作成の方法、計算機システムを利用したコミュニケーションの方法がある。授業では、主に演習により内容を体得する。

位置づけ: 4 年間の学習の前提として、計算機システムの利用方法を初歩から教授するので、前提科目はない。プログラミングや各種理論については、1 年前期「プログラミング」などに譲る。

スケジュール:

- 1 週 講義の目的、概要、計算機の利用方法
- 2 週 エディタの使用法 (基本操作)
- 3 週 ファイルの操作方法とファイルシステム
- 4 週 エディタの使用法 (仮名漢字変換)
- 5 週 電子メールの利用方法とネットワーク
- 6 週 文書作成コマンド (JLaTeX) の利用方法 (1)
- 7 週 電子掲示板の利用方法とコミュニケーション
- 8 週 エディタの使用法 (高機能コマンド 1)
- 9 週 UNIX コマンドの使用法 (1)
- 10 週 文書作成コマンド (JLaTeX) の利用方法 (2)
- 11 週 エディタの使用法 (高機能コマンド 2)
- 12 週 UNIX コマンドの使用法 (2)
- 13 週 文書作成コマンド (JLaTeX) の利用方法 (3)
- 14 週 UNIX コマンドの使用法 (3)

講義の進め方: 時間内で講義と演習とを交互に行う。課外用にはレポート課題を別に与える。課外での質疑およびレポートの提出は電子メールなどを利用して行う。

評価方法: 小テストを適宜行う。このテストの成績を 40%, 期末のテストを 30%, レポートのでき具合や教師の質問に対する解答を 30% で評価する。

教科書・参考書:

1. 九州工業大学情報科学センター (編): 利用の手引。
2. L. Lamport 著 (E. Cooke, 倉沢良一監訳): 文書処理システム LaTeX, アスキー出版局, 1990。

図 2: 平成 5 年度のシラバス

平成 5 年度は計算機リテラシーの初めての授業で、かつ、まったくの初心者に演習を行った経験が筆者にはなかったこともあり、どのようなスケジュールで、どのような内容を講義・演習すべきか雲をつかむ状態であった。しかし、色々と検討した結果、図 2 に示す要領で実施することとした。授業の目標としては、エディタ (NEmacs)、ファイル・システム、電子メール・電子掲示板、JLaTeX、各種の Unix コマンド、を習熟させることとした。これにより、“日常的”なプログラミング、コミュニケーション、レポート作成は十分に行えるはずである。

なお、教科書としては、

九州工業大学情報科学センター編: X ウィンドウによる UNIX 入門, 朝倉書店, 1993。

を使用した。図 2 とは異なっているが、このシラバスを提出した時点ではまだ書名などが決定していな

かったためである。授業は、教科書の必要か所を適宜指示しながら行うこととした。また、AV 講義室が利用できたので、講義形式と演習形式を適宜使い分けて実施した。

4 実施結果

以下では、実施した結果について、各時間ごとに紹介する。

第 1 週: 講義の目的、概要、計算機の利用方法

- 情報系の科目は、つみあげ型 → 少しでもサボるとどんどん後れる。
- ノートを取る。
- 大学は学生を大人として扱う → 指示、掲示に注意
- 分からないことはどんどん質問する → 質問がなければ分かったこととして講義が進行する → 分からないことの多くは (1) 自分の理解能力の不足ではないことが多い, (2) 他の人も分からないことが多い。
- ただし、完全に理解しなければ一歩も先に進めないようではいけない。
- 本講義では「利用方法」を示すが、計算機はそれだけではない → 理論も重要 → 逆も真 → 英会話ができないのは、英語ができないのと話す内容がないから
- 我々の目標は利用者ではなく、作成者。
- 出席を取る (finger の利用)
- 時間延長の可能性はある。

図 3: 最初に学生に指示した注意点

1 年生で計算機を直接使用して行う授業は、図 1 から分かるように、計算機リテラシーとプログラミングである。第 1 週目は、授業の曜日の関係で、学生が初めてワークステーションを使用する時間となった。2 回目以降の授業では、アシスタントとして、助手 1 名、技官 1 名、修士の TA 2 名を使ったが、最初の時間だけは、教室の他の技官 4 名にも応援を依頼し、万全を期した。

まず、講義形式で科目の目的、諸注意 (図 3) や使用する計算機の構成などを説明した。次にユーザ名とパスワードを配付し、これに関する注意をした後、X 端末に移動させた。X 端末への移動およびその構成の説明の際には、特に「キーボードにはさわらない」という指示をし、混乱を避けた。そして、いよいよ login/logout の説明および演習を行った。その際には、「間違えてもやりなおせばよい。計算機は石頭なので、聞いてきていることに答えること。」との指示も行い、学生が恐怖心を持たないように注意した。

login/logout に手間取ると予想していたが、多人数のアシスタントを使用したためもあり、案外、簡単に全学生が操作できるようになった。そこで、続いて簡単なコマンド (cal, man) や簡単なファイル操作 (cat, ls, cp, rm) を教科書に沿って説明・演習した後、NEmacs の最低限の操作方法の説明・演習を行った。

図 2 のスケジュールからすると、NEmacs の操作まで説明・演習するのは早すぎるのであるが、プログ

プログラミングの担当教官から「とにかく簡単な Pascal プログラムが書けるまで説明してほしい」との依頼があり、そのように実施した。しかし、これにはやはり無理があり、大半の学生は消化不良のまま授業が終了することとなった。

第 2 週: エディタの使用方法 (1)

第 1 週に NEmacs の操作の説明が中途半端なものとなったので、第 2 週は時間を掛けて、基本的な操作方法の説明・演習を行った。まず、文字の基本的な入力方法、カーソルの移動、Ctrl と Esc (または Meta)、表記の方法 (C-x, M-x)、M-x コマンド、コマンドとキーの対応、コマンドのキャンセル (C-g)、といった概要の説明をした。次に、起動方法やファイル名の決定と保存、終了方法をひと通り説明したあと、

cal などで作成したファイルを適当に変更せよ。ファイルの作成は各自自由にしていよい。

という指示を与えて演習を行った。

この程度は、第 1 週に少し行っており、更にプログラミングの演習でも一定量行っていたので、すぐに終了すると予想していた。しかし、実際には、中々うまく行かなかった。まず、「cal でファイルを作成せよ」が第 1 の難関であった。先週の授業内容をほとんど記憶していない学生が多数いた。次に難関であったのは「適当に変更せよ」という部分で、「適当と言われてもこまる」という雰囲気の子が多くいた。

その後、C-v や M-v C-l といったカーソルの移動、削除の演習として

% cp /home/junichi/sample/sample1.tex foo.tex で、そのファイルをながめてみよ。スペースの表記に注意。

を行った。これはこれで % を含めて入力する学生が出て、あらためて、最初は細かな部分まで指示しなければならないことが分かった。そして、最後にブロック編集と検索の説明を行ったが、十分に理解できた学生は多くなかった。

なお、最後にタイピングの練習ツールとして trr を紹介した。

第 3 週: ファイルの操作方法とファイルシステム (1)

プログラミングの演習のための最低限の操作方法是、一応、理解してもらえたとして、計算機リテラシーの本来の内容に移ることとした。まず、教科書の最初にあるハードウェアとソフトウェア、OS、UNIX を説明した後、ファイルシステムの説明を行った。

ツリー構造の話をし、ディレクトリやファイル、パス名、ファイルの保護モード、ファイル・ディレクトリの操作を教科書に沿ってひととおり説明した後、

- % cd /edu/lib/gcc-2.3.3 から始めて、cd, ls を使い、ツリー構造を調べてみよ。

- % cd で HOME にもどり, mkdir/rmdir で directory を作成・削除してみよ.
- % cd /home/junichi/sample/mode のファイルの mode と読めるかどうかを確かめよ.

という課題を与えて, 演習を行った.

学生の多くは, 完全かどうかは別として, 予定していた程度の理解を示してくれたように, その時点では見えた.

第 4 週: ファイルの操作方法とファイルシステム (2)

```
#!/bin/csh
# Online Test 93/04/26
# Jun-ichi NAKAMURA
set evalfile = /home/junichi/lec/log/t1.eval
set studentdir = t1
set file = birthday
set studentfile = ~/lec/memo/p93
echo '*****' >> $evalfile
echo 'date' >> $evalfile
echo '*****' >> $evalfile
foreach studentid ('cat $studentfile')
    echo -n $studentid
    echo -n $studentid >> $evalfile
    cd /home/$studentid
    if ( -f $studentdir/$file ) then
        echo -n "      ok"
        echo -n "      ok" >> $evalfile
        cd $studentdir
        set filewc = 'wc $file'
        set filehd = 'head -1 $file | grep 197 | wc'
        if ( $filewc[1] == 8 && $filehd[1] == 1 ) then
            echo -n "      ok"
            echo -n "      ok" >> $evalfile
        else
            echo -n "      ng"
            echo -n "      ng" >> $evalfile
        endif
        set filemd = 'ls -al'
        if ( "$filemd[3]" == "drwx--xr-x" ) then
            echo "      ok"
            echo "      ok" >> $evalfile
        else
            echo "      ng"
            echo "      ng" >> $evalfile
        endif
    else
        # file not exist
        echo "      ng      ng      ng"
        echo "      ng      ng      ng" >> $evalfile
    endif
end
```

図 4: 使用した採点用 csh プログラム

ファイルの保護モードに関しては, 理解が十分ではないようであったので, 復習のあと, 念のため,

他の学生が読めない file を適宜作成し、隣の学生同士で読めないことを確認せよ。

という演習を行った。

かなり“あぶないな”とは感じたものの、次に、

各自のホームに、t1 というディレクトリを作成し、他の学生からは読み書きできないが他人は読めるようにせよ。次にその下に、自分の生まれた月のカレンダーを cal コマンドを用いて birthday というファイル名で作成せよ。

という課題を与え、何を参照しても、他の学生と相談してもよい、という条件で“試験”を行った。図 4 に示す csh プログラムを用いて学生のファイルをスキャンし、指示どおりのファイルができていくかどうかで採点を行った。

この回は、この試験を簡単にすませた後、UNIX コマンドの紹介を行う予定であった。しかし、実際にはそれどころではなかった。課題がすぐに実行できたのは少数の学生で、かなりの学生がごく基本的な部分でつまづいていた。ファイルやディレクトリが理解できていなかったのである。

プログラミングの演習も進んでおり、学生のホームには、それなりの数のファイルが存在していた。そこで、「ファイルをまとめたもの」という簡単な説明で基本的なイメージが理解できると思っていた。しかし、ファイルそのものの理解や、「まとめる」という所でつまづいてしまう学生が少なからず存在した。

第 5 週: ファイルシステム (3), UNIX コマンド (1), 日本語入力

前回の反省から、ノートと本箱のアナログで、もう一度ファイルとディレクトリの説明を行った。やはり、“初心者のための MS-DOS 入門”といった感じの説明が必要なのである。その後、UNIX コマンドのシンタックスやファイル名の簡略指定、man コマンドを説明し、

- 計算機リテラシー用のディレクトリを作成し、そこに、プログラミング演習で作成したプログラムを cp コマンドとファイル名の簡略指定を活用して、コピーしてみよ。
- man コマンドを使って、ls や mv などのコマンドの使い方を調べてみよ。

という演習を行った。ディレクトリについては、さすがに、3 回目であるので、大半の学生は理解できたようである。

最後に、日本語入力の説明と演習を行った。仮名漢字変換はファイルやディレクトリ概念や UNIX コマンドに比べて、複雑であるので、操作できない学生が続出すると心配していた。しかし、これは取り越し苦労であり、大半の学生は、初めて楽しそうに演習を行っていた。やはり、結果がそのまま理解できるので、とっつきやすいのであろう。宿題として、

これまでの講義の関する感想を t1/kansou というファイルで作成しておくこと。

を与えて、授業を終了した。

感想としては、

- パソコンなどを使っているのに、暇である。
- 分かりやすい講義である。
- 初心者だが、何とかついていけているが、ゆっくり進んでほしい。
- 学習した操作方法の使い道が分からない。
- 一般的に道具の扱いがにがてなので、もっと丁寧に説明してほしい。
- 理解できない語句が多くて分かりにくい。
- 黒板の字をもっと分かりやすく書いてほしい。

などがあった。教官に提出する感想であることを割り引いても、好意的な意見が多かったが、理解できている学生とできていない学生の格差が非常に大きいことが気になった。

第 6 週: エディタの使用方法 (2)

ファイル操作、日本語入力がひととおり出来るようになったので、NEmacs の少し高度な使用方法の講義・演習を行った。まず、ブロック編集、検索、置換の演習を行った。演習を見ている範囲では、これらの操作の必要性が理解できない学生が少なくなかった。これは、プログラミングの演習で作成するプログラムがまだ 1 画面内に含まれる範囲であるため、このような操作が不要であるからのようである。学生には、“目の前”の課題をこなす以上のことをしたくない傾向がある。

次に NEmacs のウィンドウとバッファの説明・演習を行った。課題としては、

- C-x C-f により 2 つのファイル (たとえばプログラミングで作成したもの) を読み込み、C-x b などで比較せよ。
- ウィンドウを操作し、教科書の p.111 の図 5.3 のような画面を作ってみよ。また、それからウィンドウを 1 つだけにしてみよ。
- ウィンドウを上下に 2 分割し、それぞれに、2 つのファイルを読み込んでみよ。次に、一方のファイルの一部を別のファイルにコピーしてみよ。

を与えた。しかし、ブロック編集などと同様、これらの操作の必要性を感じないため、操作が理解しにくい学生が多かった。また、明確な指示ではなく“適当に”と言われると何をしていいのか分からなくなる傾向もあった。

第 7 週: 計算機ネットワークと電子メール

ほぼ基本的な操作の紹介が終了したので、次は電子メールの演習を行った。計算機ネットワークの説明と電子メールの操作方法、注意事項の説明のあと、mh による電子メールの発信と受信の演習を行った。

操作はかなり複雑であるので、自由に使いこなすとは言えないが、単純なメールのやりとりは、多くの学生がすぐにできるようになった。仮名漢字変換同様、使用目的のイメージが明確であり、かつ、他の学生とコミュニケーションができることが面白いためであろう。

電子メールの操作ができるようになったので、演習課題の提示もメールで行えるようになり、複雑な課題を間違いなく伝えられるようになった。そこで、

プログラミングで作成したプログラム (どれでもよい) を私 (junichi@post.isci.kyutech.ac.jp) まで、簡単なコメント (何をするプログラムかと名前) を入れてメールで送れ。

という課題で“試験”を行った。出題意図は、NEmacs のバッファ操作によりメールを書かせるものであった。しかし、学生の多くがバッファ操作が理解できていないこと、mh が多数のバッファを使うため操作がより複雑になることなどから、実行できる学生がほとんどなかった。このため、insert-file を紹介せざるをえなくなった。

表 1: メールに対する応答日時

返事のあった日			累積	備考
当日	(月)	6/07	25 (28%)	休日
+1	(火)	6/08	46 (51%)	
		6/09		
+3	(木)	6/10	64 (71%)	
+4	(金)	6/11	66 (73%)	
+7	(月)	6/14	67 (74%)	

なお、その後は、一部の学生から積極的に質問がメールで寄せられるようになった。しかし、内容の大半は、授業の性格からして当然であるが、操作方法に関するものであり、プログラミングや計算機システム、論理回路のような他の授業の教官への質問メールはほとんどなかったとのことである。ちなみに、これらの担当教官にメールで質問をしてもよいとの連絡を予告なく行い、返事を求める“実験”を行った所、表 1 に示すような結果となった。4 日以内に 73% の学生から応答があり、メールによる連絡がかなり有効に使えることが分かった。

第 8-9 週: JLaTeX, gnuplot, tgif+

第 8 週には、ワードプロセッサの機能、JLaTeX の思想・特長を説明した後、JLaTeX による文章作成の演習を行った。演習は、まず、サンプルのファイルを与え、jlatex, xdvi で処理結果を見ることから始め

た。やはり、結果が目に見えるので、学生の多くは興味を示したようである。その後、内容を簡単に説明した後、その一部を書き変える演習を行った。

第 9 週には、gnuplot と tgif+ の紹介・演習を行った。tgif+ のようなインタラクティブなアプリケーションの説明は、口頭では限度があるので、修士の TA 4 名に応援を求めて行った。JLaTeX の場合と同様、ゲーム感覚があるので、ほとんどの学生が積極的に操作を試していた。

レポートの作成に JLaTeX などを活用してくれることを期待していたが、実際には、レポートの内容、提出期限との兼ね合いの点から利用する学生は極一部であった。JLaTeX は単純な文章を作成する場合はそれほど複雑ではない。しかし、知能情報の学生が作成しなければならないレポートの多くは、プログラムの説明や論理回路の説明であり、どちらかという JLaTeX を使う積極的なメリットがないことがまず問題であった。また、プログラミングの授業が月曜の午前であったため、もし、JLaTeX でレポートを作成しようとする、情報科学センターが利用できる金曜の夜が実質的な締め切りとなった。これでは、強制するのは酷なことである。JLaTeX などを使わせるためには、他の授業との連携が、曜日の調整を含めて不可欠である。

なお、8, 9 週にネットニュースの紹介を行ったが、学生が消化不良を起こすのを避けたかったこと、情報科学センターの設定の関係で全員同時には演習ができないことから、紹介するに留めた。このため、一部の学生を除いて積極的には利用されていないようである。ただし、質問メールとその回答をニュースに流すことだけは行った。これは、同様な質問に答えることを避けたかったことと、先輩を含め、学生同士で情報交換ができるようにしたかったためである。

第 10 週: UNIX コマンド (2), ネットワーク

リダイレクション、パイプ処理、プロセスとジョブなどを教科書に沿って説明し、演習を行った。また、ネットワークの紹介、X ウィンドウシステム of 思想などの説明も行った。全員に徹底することは困難であったので、指示した課題がとにかく実行できれば、合格とした。

第 11 週: 試験

夏休み前最後の授業ということもあり、試験を実施した。試験は、オンラインで行った。問題は、図 5 に示すようなものを 10 種類 (ファイル名などを変更) 作成し、メールでの配付・答案の回収を行った。座席を指定し、付近で同じ問題が当たらないように注意した。隣と相談すること、メールを使うこととだけを禁止し、それ以外、何を参照しても、システム上のどのファイル (他の学生を含む) を参照してもよいこととした。これは、問題の趣旨からして、他の学生のファイルを参照できれば、最低限の操作ができていると考えたからである。

試験の結果は、予想していたよりも悪いものであった。かなりの学生が 30 分程度で解答を出すと考えていたが、多くの学生が時間一杯 (90 分) 使用した。また、一部ではあるが、メールの出し方がまだ分からない学生もいた。なお、JLaTeX で作成するという出題意図とは異なり、verbatim コマンドで解答を作成した学生がかなりいた。

計算機リテラシー試験問題

93/07/07

中村順一

解答を JLaTeX で作成し、そのファイルを電子メールで

`junichi@post.isci.kyutech.ac.jp`

まで送ること。その際、念のために cc により自分用のコピーを保存しておくこと。

JLaTeX は、プログラミングのレポートの形式に準拠すること。ただし、感想は不要である。

なお、採点には、正しさに加えて、解答の速さも重視する。

資料類、file などは自由に参照してよいが、口頭および電子メール、ネットニュースなどによる相談を行ってはいけない。なお、電子メールは発信者・受信者の記録が残るので、試験に関係のないことであっても、解答以外の電子メールを発信した記録があった場合は不正行為とみなす。

問題 1:

次のことを行う操作の列を示せ:

1. カレントディレクトリに `zero` というディレクトリを作成し
2. そのディレクトリの保護モードを
 - 所有者は読み、書き、実行できる、
 - 同じグループの利用者は読み、書き、実行できる、
 - 他人は読み、書き、実行できない、
 ようにする。
3. カレントディレクトリにある `one.p` を `zero` の下にコピーする。

問題 2:

次の Unix コマンドの意味と実際の実行例（結果が分かるように）を示せ。ただし、例は、教科書のものとは異なること。

```
% ls
```

問題 3:

次のコマンドの動作の意味を説明せよ。

```
% ls | tr a-e A-E > two
```

問題 4:

NEmacs のみを用いて `three.p` というプログラムの一部を変更し、`four.p` というプログラムにする場合の操作の概略を示せ。

図 5: 試験問題の例

第 12 週: 復習, UNIX コマンド (3)

夏休み明けでもあり、NEmacs やメールの復習をまず行い、`who`, `finger`, `grep`, `sort`, `uniq`, `wc`, `awk` などの説明と以下のような簡単な演習を行った。

- これまでに作成したパスカル・プログラムで、ある変数（適当でよい）を使っているプログラムを見つけ出してみよ。
- `begin` を何回使っているかを数えてみよ。
- 今 login している計算機で、すべてで幾つの `kterm` が実行されているかを数えてみよ。

夏休みによるギャップは予想していたよりも少なかったようである。

第 13 週: シェルプログラミング入門

授業の最後として、シェルプログラミングの演習を行った。

- 教科書のサンプルを、コンパイルに失敗した場合には NEmacs を呼び出すように変更してみよ。
- 複数のパスカル・プログラムを次々にコンパイル・実行するシェルプログラムを作成せよ。

という課題を与えた。すでに Pascal プログラミングが進んでいるためか、かなりの学生が実行できた。

5 Q & A

ある学外の方から、計算機リテラシの授業に関していくつかの質問を頂いた。以下に質問の要旨とその回答の一部を示す。

1. エディタは何を使っているか。

nemacs です。ただし、beginner mode というものを作り、「危険なコマンド」たとえば、C-q とか C-h とかを使えなくしています。基本的にはモードが変化するコマンドを使えなくしてます。何か操作していて C-h を入れてしまって、パニックをおこすのを防ぐ意味です。もちろんこれは、.emacs の初期値に入れてあるので、学生が解除することができます。

2. jvi を採用しなかった理由は。

emacs か vi かはあまり検討しませんでした。というか、vi はほとんど考えてなかった。

3. 授業をする上で nemacs では困ったというようなことがあったか。

上記のように Beginner mode を作ったので、特に困ったことはありません。また、CPU パワーについては、機種選定の段階から nemacs の稼働に問題がないことを条件にしていたので、こちらも問題なかったようです。

ただし、nemacs の buffer と window の関係については、あまり理解してもらえなかったようです。

4. 電子メールを使う際のモラル等についてどのくらい教えているか。

電子メールのモラルは最初は教えようと思っていたのですが、学生の能力や時間の関係で、簡単にふれた程度です。また、news については、CPU パワーの関係や演習の支障になるとの点から起動を制限しています。このため、「さあ、みなさんやってみましょう」というのができなかったのも、紹

介するにとどめました。なお、default では学内 news group のみにしています。(専門になると all)。

5. 文書作成システムとして JLaTeX を採用しているようであるが、図を描くためのソフト Tgif 等を教えたか。

はい、教えました。あと、X-window 入門には gnuplot の説明もあるので、これも紹介しました。

しかし、実際には学生はあまり理解してくれなかったようです。どうも大半の学生は興味を持たないようで、単位とは関係のないことはしようとしません。他の科目(プログラミングや計算機システム)のレポートを JLaTeX で出すように担当者に依頼して、そのようにしてもらったんですが、ほとんどの学生は「お茶をにごす」程度だったようです。

ちなみに、私のレポート・試験課題として、問題の答えを JLaTeX 形式で出せというと、verbatim でやってくる学生が多数いました。

6. タイピングの練習はしているか。

trr を紹介して、各自練習するように、ということはしましたが、特に授業中には時間は取りませんでした。学生を見ている感じでは、高速なタイピングは別として、1 本指の人はいないようでした。

7. 「計算機リテラシィ」では、簡単なシェルプログラムや C 言語などのことは教えているか。

csh についてはリテラシィでやりましたが、一般のプログラミングは、他に演習付き授業があるので、やっていません。カリキュラムの改訂でリテラシィを導入したのは、プログラミングの授業の片手間ではできないから、というのが大きな理由です。

8. 演習はすべて演習室で行っているのか。それとも、前半だけ講義室で説明し、そのあと演習というスタイルか。

部屋の中央部分で講義ができ、周辺に X-terminal がある、という部屋をつかっていましたので、適宜、講義と演習をまぜました。もっとも、完全な講義スタイルはせずに、演習スタイルが中心でした。

9. 「計算機リテラシィ」では、何人のスタッフで演習を行っているか(教官、助手、TA など人数)。

私と助手 1 名、技官 1 名(結構ハッカー的な人ですが)、TA 2 名で通常はしていました。第 1 回目(login させる時)は、学科の他の技官 4 名にも手伝ってもらいました。また、Tgif の演習の時は、あと何人かの M の学生にボランティアで手伝ってもらいました。

10. 演習中に分からないことがあった時、学生は、おもにどのようにして解決しているか。(自力で、友人に聞いて、先生に聞いて)。

原則的には、近くにいる助手、技官、TA に聞いてもらってますが、友人というのも多いですね。たしかに、一部にスキモノがいますから、そのような学生を活用する方法を考えた方がいいなと思っています。

あと、メールによる質問を積極的に受け付けました。特に LaTeX なんかはよく質問が来ました。それに、その質問を news に流しました。場合によっては、別の学生 (同年とは限らず、修士の学生なんかを含む) が答えてくれました。

11. 学生がパソコンをまったく知らないと仮定して授業を行うと、最初から個人差が激しく、講義の進め方がむずかしくなるのではないか。

個人差は確かに問題です。しかし、何も知らない学生を対象に講義を進めました。良く分かっている学生は、かってにやらせておきました。また、演習中に目についたまあ分かっている学生には、マンツーマンで、おもしろいコマンドの紹介なんかをしました。またメールでの質問に答える、というのもしました。

6 おわりに

本稿では、計算機の“操作方法”を教えるという従来はなかった授業を実施した経験を紹介した。初めての講義ということもあり、うまくいかなかった点多かったが、まずまず成功したと考えている。今回の経験から平成 6 年度は、以下のことを念頭に置いて授業を組み立てる予定である。

- 最初に計算機の利用全体のイメージを与える。UNIX の利用方法を紹介したビデオを利用する。
- メール、tgif+ といった“遊び”に近いものを前半に、ファイル・システムや UNIX コマンドのような初心者の学生にとって抽象的なものは後半に、説明・演習を行う。
- 計算機を操作する時間を多くするため、JLaTeX によるレポート提出をできるだけ強制する。
- よく分かっている学生とよくわからない学生が混在している点に注意する。