

UNIXシステムコールの教育

3S-5

稚内北星短大 姫宮利融

1 緒言

稚内北星短大では、開学以来、2年次の情報教育の中でUNIXシステムコールの教育を行い4年間の教育実践を重ねてきた。そこで、UNIXシステムコールの教育の内容、成果、問題点をまとめる。

2 UNIXシステムコールの教育の目的

当短大でUNIXシステムコールの教育を行ってきたのは、1つはオペレーティング・システムを理解する上で、UNIXシステムコールの教育が有効な方法だからである。もちろん、その前に、コンピュータの利用の上でOSとしてUNIXを使う機会が増え、様々なUNIX上のシステムプログラミングを行うことが必要になっていることは言うまでもない。とはいえ、オペレーティング・システムの仕組みを理解することは、いわゆるアプリケーション・プログラムを専ら行なうプログラマーにとっても必要になっていると考える。

OSの理解という点では、第1に、catやcp(copy)などの基本的なコマンド(ツール)の仕組みを理解させること、第2に、それと表裏の関係にあるのだがUNIXのファイルシステムの特徴を理解させること、第3に、シェルによるプログラム実行の仕組みを理解させることに置いた。

UNIXのシステムコールの教育の中では、もう1つ、ネットワーク(プロセス間通信)を扱い、Berkley Socketによってプログラミングを行い通信を実際に体験させることを行ってきた。

3 教育内容の概要と教育方法

当短大では1年次にUNIXの利用(シェルプログラミングまで)とC言語によるプログラミングを教育している。その上になつて、2年次のシステムコールの教育では、

1. 低レベル入出力—FILE I/Oとファイルシステム、
2. プロセスの制御—forkとexec、
3. パイプの仕組みとパイプラインの作り方—pipeとdup、

4. socketの概要

を1年間をかけて教育を行った。

授業は毎週1コマの講義とそれに対応する1コマの実習からなり、講義でそれぞれのシステムコールの意義と具体的な利用法、設計上の考え方を説明し、実習で例題の実行と応用課題を解くことを行わせた。実習は53台のワークステーションをネットワークで結んだ実習室で学生1人に1台の環境で行われた。この環境によってプログラミングの作業効率が向上し、プロセスの自由な制御が容易になり、プロセス間通信の体験が手近なものとなった。

以下、それぞれの項目について特徴的なことを述べる。

4 FILE I/O

教材として、catの私家版やcpの私家版を取り上げてUNIXの基本的なコマンド(ツール)をプログラミングしてみることから行った。その中で、低レベル入出力関数と標準入出力関数の違いに留意した。また、creatシステムコールとopenシステムコール(オプション・フラグつき)の違いとオプション・フラグの使い方をコピー・ツールを何種類か作ることによって示した。(Table)ファイルシステムの説明ではiノードとディレクトリの説明を行い、stat, fstatシステムコールを使ってプログラミングする例を示した。

Table Copy Tools

prog.	コピー先ファイルの作成やオープン
mycp	creat(path, 0644)
mycp2	open(path, 0644)
mycp3	open(path, O_CREAT   O_EXCL, 0644) open(path, O_TRUNC); open(path, O_WRONLY) 対話式
mycp4	stat(oldfile, &sti) creat(path, sti.st_mode)

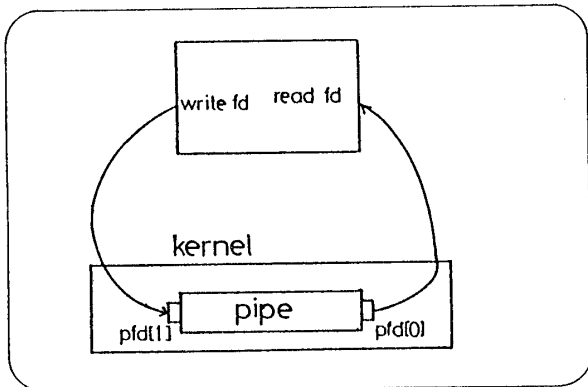
5 プロセス—forkとexec

execとfork(とwait)のシステムコールを教え、「シェルもどき」(環境を自前で更新せず、プロンプトを出して単純なコマンドのみを実行するもの)を示してシェルの基本的な仕組みを教えた。また、環境を自前で更新するもの([1]に載っている)を説明して実行させプログラムとしてのシェルが環境変数を自前で設定する意義を体験させた。

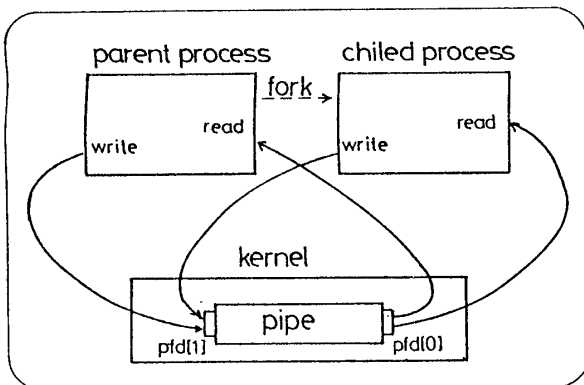
## 6 pipe と dup

pipe システムコールと fork の関係を図式的に説明し (Fig.1)、dup の説明と併せてパイプラインを構成する方法を説明した。また、dup を使ってリダイレクションと等価なことを実行する方法を説明した。

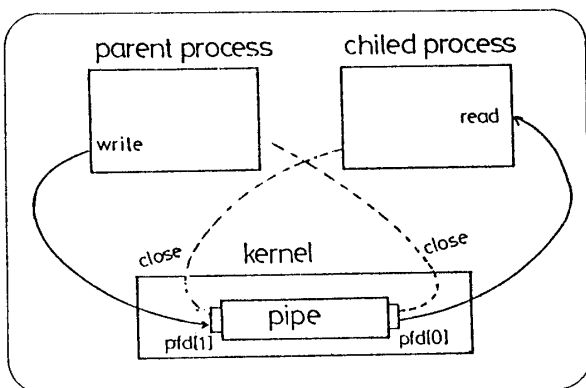
親が書いて子が読むこと



単一のプロセスでのパイプ



フォークの直後



余分な書き込み端・読みだし端をクローズ

Fig.1 Schematic diagrams of using "pipe"

## 7 ネットワーク

本年4月に出来た実習室は1人1台のワークステーションという環境なので、ネットワークの教育は機材の上できわめて容易になった。教育で取り上げたのは、主としてTCPによる通信であるが、1台のワークステーション上の異なるウィンドウ間でそれぞれの学生がclient-serverの通信を行う実験が出来るので、それぞれの学生に自己充足的に実験させて次に異なるホストの間の通信に進んだ。

## 8 教育の到達点の評価と問題点

12月はじめの中間試験の成績のヒストグラムを Fig.2 に示す。これは、シェルで自前で環境変数を更新する意義を問うもの1題と机上でプログラミングしてパイプラインとリダイレクションを実現する問題が1問ずつで、すべて記述式である。成績上位の層がそれなりに形成されていて、オペレーティング・システムの理解という点でそれなりの成果を挙げている。もちろん、下位の層も多い。

UNIXシステムコールの理解の上で学生が感じている困難点は、およそ次の点にあると思われる。

1. Cの文法と作法の理解の不十分さ。特に混乱しやすいのは構造体へのポインタの使い方や  

```
if(systemcall()==-1) error();
```

と書くようなエラーチェックの慣習
2. 関数の呼び出しとそれからの戻り値、引数による値の引渡し、
3. システムコールはカーネルに対する命令であり、カーネルの働きがある程度分からないとシステムコールの意義が分からないこと。

Reference [1] Marc J. Rochkind (福崎俊博訳) 「UNIXシステムコール・プログラミング」(アスキー出版局)

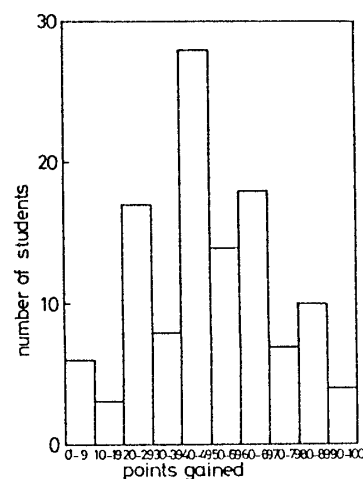


Fig.2 Results of EXAMINATION