

ジョブ

1 マルチプロセス

UNIX はマルチプロセス機構によって、複数のプログラム(プロセス)を同時に実行できる。実際には同時ではなくタイムシェアリングにより細かく処理を切り替えている。この技術により複数の人が同時に計算機を使用したり、プログラムを開発しながら同時にドキュメントを作成したりすることができる。

2 フォアグラウンドジョブ

tcsh などのシェルは、コマンドなどからプログラムを実行する。この実行を開始したプログラムのことをジョブ (job) と呼ぶ。このプログラムが終了するまでシェルではコマンドプロンプトが表示されない。この状態をジョブがフォアグラウンドで実行されていると呼ぶ。ジョブを中断するときは Ctr-z を押せば良い。そうするとジョブは中断されコマンドプロンプトが表示される。Ctr-c でフォアグラウンドのジョブは中止される

3 バックグラウンドジョブ

シェルでプログラムを実行しながらさらに次のプログラムを実行することができる状態にすることができる。このことをジョブがバックグラウンドで実行されていると言い、コマンド入力の際に & を入力すればよい。

```
kubo@kago[~]%netscape &
```

こうすると続いてコマンドを入力できる。以下にジョブに関するコマンドを書いている。%1 となっているのはジョブ番号である。

- bg: バックグラウンドのジョブを表示させる
- fg: 中断中またはバックグラウンドのジョブをフォアグラウンドで実行
fg %1
- jobs: ジョブの番号と状態を表示
- kill: バックグラウンドジョブを中止
kill %1

4 ジョブとプロセス

シェルで実行しているプログラムのことであり、同じシェルで実行しているジョブしか jobs コマンドは表示しない。現在実行中のすべてのプロセスは ps コマンドによって表示でき kill コマンドにより中止した

り再起動したりすることができる。

ps コマンドのオプションを以下に書く。

- a:すべてのユーザのプロセスについて表示
- u:プロセス情報を詳細に表示
- x:ps コマンドを実行した端末以外の端末に入出力を行うプロセスについても表示
- w:コマンド名をすべて表示

良く使う形式は

```
kubo@kago[~]%ps aux
```

ps コマンドでプロセス番号がわかれば強制的に終了させたり再起動させたりすることができる。

```
kubo@kago[~]% ps aux | grep xmms
kubo  5125  3.9  4.0  9476 4988  p1  S    5:26PM  0:00.78 xmms
kubo@kago[~]% kill 5125
```

```
[2]  Terminated                xmms
```

オプション HUP を使うと再起動する。

```
kago# ps aux | grep inetd
root  188  0.0  0.6  1048  724  ??  Is   Mon07PM  0:00.03 inetd -wW
kago# kill -HUP 188
```

どうしても終了しないプロセスはオプション 9 を使うと終了できることもある。

```
kubo@kago[~]% ps aux | grep xmms
kubo  5128  3.1  4.2  9476 5200  p1  S    5:28PM  0:00.82 xmms
kubo@kago[~]% kill -9 5128
```

```
[2]  Killed                          xmms
```