

UNIX コンピューター操作法の最終回として、プロセスの概念と、ネットワークを介して離れた場所にあるコンピュータを利用する方法について学ぶ。

4. UNIX マシンでは複数の処理を同時並行で実行可能

UNIX はネットワークに接続されたサーバー用の OS として利用されてきた歴史があるため、1 台のマシンを複数の人が同時に使用するための仕掛けが豊富である。単にユーザーをユーザー名とパスワードで管理しているだけでなく、同時に複数のユーザーが 1 台のマシンにログインしてコンピュータの処理能力を共用することができる。

複数ユーザーが同時に利用しても支障が無いように、UNIX は(LINUX も)、複数の命令を同時並行で処理する¹機能を備えている。この機能のおかげで、一人のユーザーが複数の処理(1つ1つの処理のことをプロセスという)を同時並行で実行させることも可能となっている。まずは関連する概念と用語を説明する。

4.a. プロセス

端末ウィンドウを 1 つ開き、次のコマンドを実行してみよう。

```
ps ↵
```

(実行中のプロセスの一覧を表示)

ps は、プロセスの一覧を表示するコマンドだ。次の図のような結果が返ってくるだろう。今実行した ps のほかに、シェルである tcsh がリストされている。

ps の結果の例

PID	TTY	TIME	CMD
3287	pts/1	00:00:00	tcsh
3927	pts/1	00:00:00	ps

実行中のプロセスには全てプロセス ID という番号が割り振られている。ps の結果の「PID」の欄に表示されている数字がプロセス ID だ。

ps コマンドは、デフォルト²では、自分がその端末ウィンドウで実行しているプロセスしか表示しない。UNIX パソコンでは、常にたくさんのプロセスが動いている。画面にウィンドウを表示するためのプロセスやネットワークサービスを提供するためのプロセスだ。

1 このことをマルチタスクという。

2 「デフォルト(default) (省略時設定ともいう)とは、あらかじめセットされている標準的な設定のこと。

それら全ての一覧を表示するには次のようにする。

```
ps -ef ↵
```

(そのコンピュータで実行されている全てのプロセスの一覧を表示)

上のコマンドを実際に実行すると、多数のプロセスが表示される。しかし、それらの多くは待機中のプロセスであり、マシンの処理能力はさほど消費されていない。処理能力に対する負荷の大きい順にプロセスをリストアップするコマンドが **top** である。新しい端末ウィンドウを開いて、次のコマンドを実行してみよう。

```
top ↵
```

(負荷の大きいプロセスを監視)

実験として、何の役にも立たない計算を無限に繰り返すプログラム³を実行してみよう。プログラムのファイル名は **endless.pl** である。top を実行していない端末ウィンドウで、次のようにタイプする⁴。top の画面がどのように変化するか確かめよ。endless.pl は待っていても終了しない。終了方法は後で述べる(4.b)。

```
./endless.pl ↵
```

4.b. バックグラウンドジョブ

endless.pl を実行中の端末ウィンドウでは他のコマンドを実行できなくなっていることに気づいたと思う。このように、実行中のプログラムが端末を占領している場合、そのプログラムはフォアグラウンドで実行されている、という。

1つの端末ウィンドウで、複数のプロセスを動かしたいときには、プログラムをバックグラウンドで実行すればよい。方法は簡単で、実行するときのコマンド文字列の末尾に **&** を付ける。例えば、endless.pl をバックグラウンドで実行するには、下のようになれば良い。新しい端末ウィンドウを開き、実行してみよう。

```
./endless.pl & ↵
```

(endless.pl をバックグラウンドで実行)

こうすると、実行開始後にプロンプトが表示される(つまり、次のコマンドを入力できる)

3 配布する。コピーする方法は実習時に指示する。

4 そのまゝに、chmod で実行可能にする必要があるかもしれない。

ことを確認しよう。また、**top** の表示がどうなるかも見てみよう。さらに、**endless.pl** を複数個バックグラウンドで実行したらどうなるか試してみよ。

4.b. フォアグラウンドで実行中の処理を強制終了する方法

フォアグラウンドで実行中のコマンドやプログラムが、正常な方法で終了させられなくなったときは、**「Ctrl」+ c** を押す (**「Ctrl」** を押しながら **c** を押す)。プログラムが無限ループに陥ったと思われるときにはこの方法で止めればよい。

4.c. 4.b の方法ではだめなときの方法 (プロセスを殺す)

プロセス ID が必要になる場面の 1 つは、実行中に 制御不能に陥ったプログラムを強制的に止めたい場合 である。

kill コマンドは、指定されたプロセスを終了させる⁵。下の構文を参考にして、さっき起動した **endless.pl** を殺してみよう。

```
kill プロセス ID ↵  
(指定されたプロセスを殺す)
```

コンピューターを利用していると、プログラムがどうしてもなく暴走してしまい、上述のいずれの方法でも殺せないケースにぶつかることがある。そのような場合は、**-9** オプションを付けて **kill** を実行してみる。

```
kill -9 プロセス ID ↵  
(必殺技)
```

それでも殺せないケースが発生した時には、教員に申し出なさい。

4.d. プログラムを一時停止させる方法

実行中のプログラムを一時停止させるには、**Ctrl-z** (**「Ctrl」** を押しながら **z** を押す) を押す。プログラムがフォアグラウンドで走っていても、停止させてしまえば、同じ端末にプロンプトが表れ、別のプログラムの実行が可能となる。

4.e. 停止させたプログラムを再開する方法

一時停止させたプログラムをフォアグラウンドで再開するには、**fg** コマンドを使う。

⁵ もちろん、他のユーザーのプロセスを勝手に殺すことはできないようになっている。また、**kill** は、プロセスを殺す以外の用途もある。知りたい者は **man kill** を参照のこと。

fg ↵

一時停止させたプログラムをバックグラウンドで再開するには、**bg** コマンドを使う。

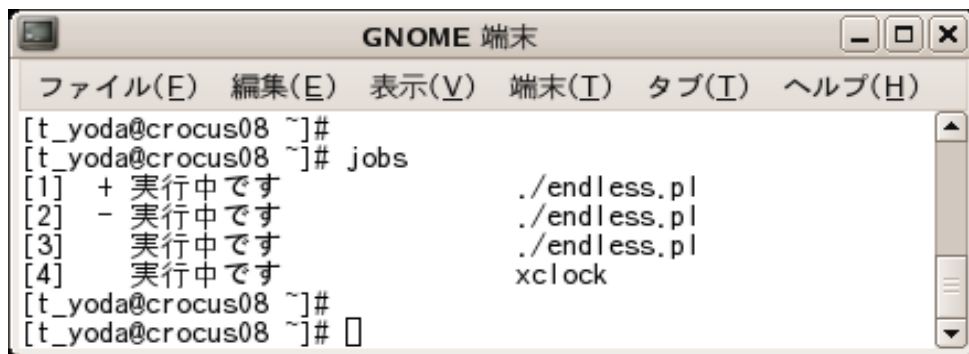
bg ↵

バックグラウンドで実行中のプログラムを一時停止させる最も単純な方法は、いったんフォアグラウンドへ移行させてから **Ctrl-Z** を押すことだ。バックグラウンドで実行中のプログラムをフォアグラウンドへ移行させるためには、**fg** コマンドを次の構文で使う。

fg %ジョブ番号 ↵

ジョブ番号とプロセス ID は別物である。ジョブ番号を知るには、(そのプログラムを起動した端末で) **jobs** コマンドを使う。⁶

最左列の[]に
囲まれた数字
(1 とか 2 と
か)がジョブ
番号



5. UNIX マシンは遠隔操作が可能

既に述べたように、遠隔地にいる多数のユーザーがネットワークを介して利用するサーバーの多くは **OS** として **UNIX** や **LINUX** を採用している。実習室の **LINUX** マシンも、他のマシンからネットワークを介してログインして利用できる。

ssh (「エスエスエイチ」と読む) というコマンドを使うと、インターネットを介して他のマシンにログインすることができる⁷。

例えば、1000 回の **blast** 検索を行わなければならないときに **ssh** を利用して 2 台のマシンで実行すれば、パワーが 2 倍になるので所要時間は半分になる。このように、離れたとこ

6 ジョブとプロセスの違いは説明しない。知りたい者は適当な参考書を読むか、個別に教員に質問しなさい。

7 もちろん、相手のマシンを利用するためのユーザー登録とパスワードが必要だ。

ろにある(=リモートの)マシンのパワーを、普段の仕事場から一歩も離れることなく利用することができる⁸。

では、リモートのコンピュータにログインして利用する方法と、リモートのコンピュータとファイルをやりとりする方法を練習しよう。ssh コマンドの書式は次の通り。

```
ssh オプション ユーザー名@リモートのマシン名 ↵
```

ただし、リモートのマシンにおける自分のユーザー名が、手元のマシンにおける自分のユーザー名と同じならば、ユーザー名@は省略可能。

例えば、実習室の他のマシンに入るには、次のようにすれば良い⁹。

```
ssh -X リモートマシンの名前(IPPC107 とか) ↵
```

試しに、隣のマシンに ssh を使って接続してみよう。また、今現在そのコンピュータを利用している人の一覧を知るコマンドである who を実行して、隣の人がログインしたことを確認してみよう。

```
who ↵
```

さらに、接続した先(となりのマシン)で、endless.pl を実行してみよう。隣の人は、top の画面がどのように変化するかを確かめてみよう。

インターネットを介して、遠く離れた場所¹⁰にあるコンピュータに入ることもできる。例えば、次のようにする。¹¹

```
ssh tanaka@eniac.nagahana-i-bio.ac.jp ↵
```

6. 遠隔のコンピュータとファイルの送受信を行うためのコマンド

リモートのマシンからファイルをダウンロードしたり、リモートマシンにファイルをアップロードしたりするには、通常、sftp というコマンドを使う。この実習中に sftp を使う必要に迫られる場面は無いと思われるが、コンピュータに関する一般常識なので、使用法を簡単に説明しておく。詳しく知りたい人は、参考書を見て下さい。

8 ゆえに運動不足になりやすい。休み時間などに適度に体を動かそう。

9 -X オプションをつけると、リモートで実行した X のソフトの画面を、手元に出すことができるようになる。

10 海外のコンピュータに接続することも可能。

11 eniac.nagahana-i-bio.ac.jp は架空のマシン名

1. まず、次のようなコマンドを実行して、手元のマシンとリモートのマシンとの接続を行う。「ユーザー名」は、リモートのマシンにおける自分のユーザー名。

```
sftp ユーザー名@リモートのマシン名 ↵
```

リモートマシン用のパスワードの入力を求められたら、入力する。接続に成功すると、**sftp>** のようなプロンプトが表示される。

2. (リモートの)ホームディレクトリが(リモートの)カレントディレクトリになる。必要に応じて他のディレクトリに移動する。**cd** や **pwd** コマンドが使えるが、機能はやや制限される。
3. **ls** コマンドを実行すると、ファイルのリストが表示される。
4. リモートのファイルを手元のコンピュータに転送するには、(**sftp>** のプロンプトに対して) **get** コマンドを入力する。書式は次の通り。

```
get ファイル名 ↵
```

「ファイル名」には、ワイルドカード(*)も使える。

5. 逆に、手元のマシンに置いてあるファイルをリモートのマシンに転送するには、(**sftp>** のプロンプトに対して) **put** コマンドを次の書式で入力する。

```
put ファイル名 ↵
```

6. 用事がすんで、リモートマシンとの接続を切るには、**exit** コマンドを入力する。