

5.6. ジョブコントロール

Linux は複数のプログラム (プロセス) を同時に動かすことができる、マルチタスキングをサポートしています。しかし、端末 (GUI であれば GNOME 端末等) で動かしている場合に、一つの端末で同時に一つのプロセスしか扱えないというのは不便です。そこで、ここではジョブコントロールという操作について紹介したいと思います。

ただし、端末操作を行うプログラムをジョブコントロールで操作すると画面表示が乱れる、「表示待ち」状態に入って一時停止してしまうなどの問題もあります。このようなプロセスを使う時にはあまり無理にコントロールしないほうがいいでしょう。

ここでは練習として、`sleep` コマンドを使ってみましょう。このコマンドは、指定した時間だけ待つというだけのコマンドです。単体で使用することはあまりなく、プログラムを実行する際に時間調整として用いることがほとんどです。たとえば、

```
$ sleep 10
$ ←10 秒後にプロンプトが戻る
```

という感じで動作します。

まずはバックグラウンド実行を紹介します。この操作を行うためにはコマンドの最後に“&”を追加するだけです。バックグラウンド実行すると、即座にプロンプトが戻ってきます。

```
$ sleep 10 &
[1] 29935
$
[1]+  終了                sleep 10
```

この時に画面に表示されるのは、ジョブ番号 (1) とプロセス ID (29935) です。プロセス ID については後述します。この場合、ジョブ番号 1 として動くことになったコマンドは、10 秒後に終了するので、10 秒以上経過した状態でプロンプトの書き換え (コマンドを実行してプロンプトが表示されるタイミング等) が発生すると、通知が表示されます。

次にキー操作として“Ctrl+z”を紹介します。この操作を行うと、実行中のプロセスが停止状態になります。

```
$ sleep 10
^Z[2]  終了                sleep 10

[3]+  停止                sleep 10
$
```

プロンプトが出ていますから、このまま別のコマンドを実行することができますが停止させたジョブはずっと停止したままです。この状態で2つの操作を行うことができます。

- **bg** このコマンドは入力すると、止めていたプロセスをバックグラウンドで再開することができます。結果的に“&”をつけて実行するのと同じになります。
- **fg** このコマンドを入力すると、止めていたプロセスがフォアグラウンドで再復帰して実行を再開します。(フォアグラウンドで再開した場合、そのプロセスを停止もしくは終了させるまでプロンプトは戻ってこない)。

ここでは“fg”にしてみましょう

```
$ fg
sleep 10
```

複数のプログラムをジョブコントロールしている場合や、どのプログラムが待機状態になっているのかを忘れてしまった場合には、`jobs` コマンドで確認することができます。

```
$ jobs
[1]+  Stopped                  sleep 10
```

停止中のプロセスがある場合には、ジョブ番号を指定することで、特定のジョブを指示して制御することが可能です。

この場合のジョブ番号の指定は“%ジョブ番号”という記述になりますので注意して下さい。なお、バックグラウンド動作中のプロセスは、一部の例外を除き、**端末終了時に一緒に終了してしまう**ので、重要な作業をバックグラウンドで走らせている場合には注意して下さい。

```
$ bg %1
[1]+ sleep 10 &
$ jobs
[1]+  Done                    sleep 10
[1]+  Stopped                  sleep 10
```