

UNIXサーバー構築Ⅱ

第8章 サーバー運用・管理Ⅰ

プロセス管理

プロセスを管理するために、**psコマンド**を使用する。オプションとして**aux**を使用することが多い。

■プロセス

メモリが割り当てられ、CPUによって実行中のプログラム

sudo ps aux ... システム上で実行されているすべてのプロセス

〈例〉sshのプロセスのみを表示

sudo ps aux | grep ssh

■psコマンドの主な出力項目

USER	: プロセスを実行しているユーザー
PID	: プロセスID
%CPU	: CPUの利用率
%MEM	: メモリ利用率
TTY	: プロセスを実行した端末名
STAT	: プロセスの状態
START	: プロセスの実行日時
TIME	: 消費したCPU時間
COMMAND	: コマンドまたはプログラム名

■ pstree

プロセスの親子関係を表示

sudo pstree

■ プロセスの制御

プログラムの実行が終了すると、プロセスは消滅する

サーバーのプログラムや異常動作を起こして

終了しなくなった場合にプロセスを制御する必要がある

- ・プロセスを制御する信号のことを**シグナル**と呼ぶ
- ・プロセスを制御する＝プロセスに**シグナル**を送る

■ プロセス制御

kill [-[SIG]シグナル] PID

プロセスに**シグナル**を送るコマンド

■ 主なシグナル (kill -l コマンドで確認できる)

シグナル名	シグナル番号	説明
HUP	1	ハングアップ
KILL	9	強制終了
TERM	15	終了
CONT	18	プロセスの再開
STOP	19	一時停止

プロセス名でプロセスを制御

killall プロセス名

同じ名前のプロセスが、複数実行されている場合は、**すべてにシグナルが送られてしまう**ので注意。

■topコマンド

システムの状態を**継続的**に監視・確認することができるコマンド。

CPUやメモリの使用量、システムの負荷状態などを監視することができる。

top オプション

・オプション

-b(バッチモードで実行)、-n 回数(更新する回数を指定)

-d 秒(更新する間隔を指定)、-u ユーザ名(指定したユーザのみ)

シェル環境

■ジョブ

ユーザーがコマンドやプロセスをシェル上で実行するまとまりの単位。フォアグラウンドまたは、バックグラウンドで実行される。

■フォアグラウンド

実行中のジョブが終了するまで、シェル上での作業はできない

■バックグラウンド

ジョブの終了を待たずにシェルを使うことができる

ジョブの実行モード

& コマンドラインの最後に「&」を付けるとバックグラウンドで起動

jobs 実行中のバックグラウンドジョブを表示

fg バックグラウンドジョブをフォアグラウンドで実行する

Ctrl+Z 「一時的にジョブを停止」(中断)

bg フォアグラウンドジョブをバックグラウンドで実行する

その他コマンド

- 指定した時間だけ処理を遅延(シェルを停止)する

sudo sleep

<例>60秒間処理を遅延する

```
sudo sleep 60
```

ディスク管理

■dfコマンド

ディスクの空容量を確認する。

df オプション

・オプション

-h サイズに応じて読みやすい単位で表示

-T ファイルシステムのタイプなどを表示

-H 読みやすい単位で表示

その他に-k,-mなどのオプションもある。

ディスク管理

■duコマンド

ディスクの使用量を確認する。

du ファイル名(ディレクトリ名) オプション

・オプション

- s サブディレクトリのサイズを含めない
 - s 指定したディレクトリの合計のみ表示
 - d 深さ 集計するディレクトリの深さを指定する
- その他に-p,-Lなどのオプションもある。

ディスク管理

■ fsckコマンド

ファイルシステムのチェックと修正を行う。

fsck オプション デバイス名

・オプション

-l パーティションのリスト表示

-t 種類 ファイルシステムの種類の指定

-A /etc/fstabにあるファイルシステムを全て検査する

その他に-v,-cなどのオプションもある。

メモリ管理

■freeコマンド

メモリの使用状況を確認する。

free オプション

・オプション

-h 読みやすい単位で表示

-b バイト単位で表示

-t 物理メモリとスワップメモリの合計を示す行も表示する

その他に-k,-m,-gなどのオプションもある。

cron

■cronとは

cronとは、定期的にプログラム(シェルスクリプトなど)を実行するための機能。Linuxにはデフォルトでインストールされている。また、cronはユーザごとに設定ができる。

■Cronの設定

crontabコマンドや/etc/cron.d/ファイルを編集する方法がある。

crontabコマンドの使用は現在非推奨で、/etc/cron.d/ファイルを使用することが多い。

/etc/cron.d/ファイルによる設定

/etc/cron.d/ディレクトリにファイルを作成し、cronの設定を行う。

①ファイルをコピー

cronの設定ファイルをコピーする。コピーせずに作成することも可能。

sudo cp /etc/crontab /etc/cron.d/cron_test(拡張子はなし)

②設定ファイルを編集

sudo vi /etc/cron.d/cron_test

ファイル構文 ... 分 時 日 月 曜日 実行するユーザー名 コマンド

〈例〉設定例

59 23 * * * **userA** /bin/bash /home/userA/test/**test.sh**

* 毎日**23時59分**に**userA**(実行するユーザー)がtest.shを実行する。

* /5 * * * * userA /bin/bash /home/ubuntu/test.sh

* 5分ごとにtest.shを実行する。

③cronの再起動

sudo systemctl restart cron

* sudo systemctl status cron ... 状態を確認

crontab

■ Crontabコマンドを使用

① タスクスケジュールの確認(ユーザごとの確認)

crontab -l

* cronの設定がない場合「**no crontab for** ie2a99」のようなメッセージが表示される。

② スケジュールの設定

crontab -e

編集するエディタを選び、ファイルを修正する。

構文 … 分 時 日 月 曜日(0～7) コマンド

■cronの削除

crontab -r

■cronのログ管理

cronのログは/var/log/syslogに保存される。

cat /var/log/syslog | grep CRON

less /var/log/syslog | grep CRON