オペレーティングシステム演習

第11回（2009.06.25）

プロセスとメモリ管理

プログラムとプロセス

- プログラム
  - コンピュータを動作させる命令語とデータ
  - 実行していない時も、コンピュータ上にファイルとして存在している。
- コマンドもプログラム
  - /usr/binのディレクトリ
  - MINIXの「コマンド」は、ほとんどプログラムとして提供されている。
    - シェルが処理しているものもある。
- プログラムが実行されると、それが「プロセス」になる。
  - プログラムの実行が終了すると、プロセスも終了する。

プログラムの実行

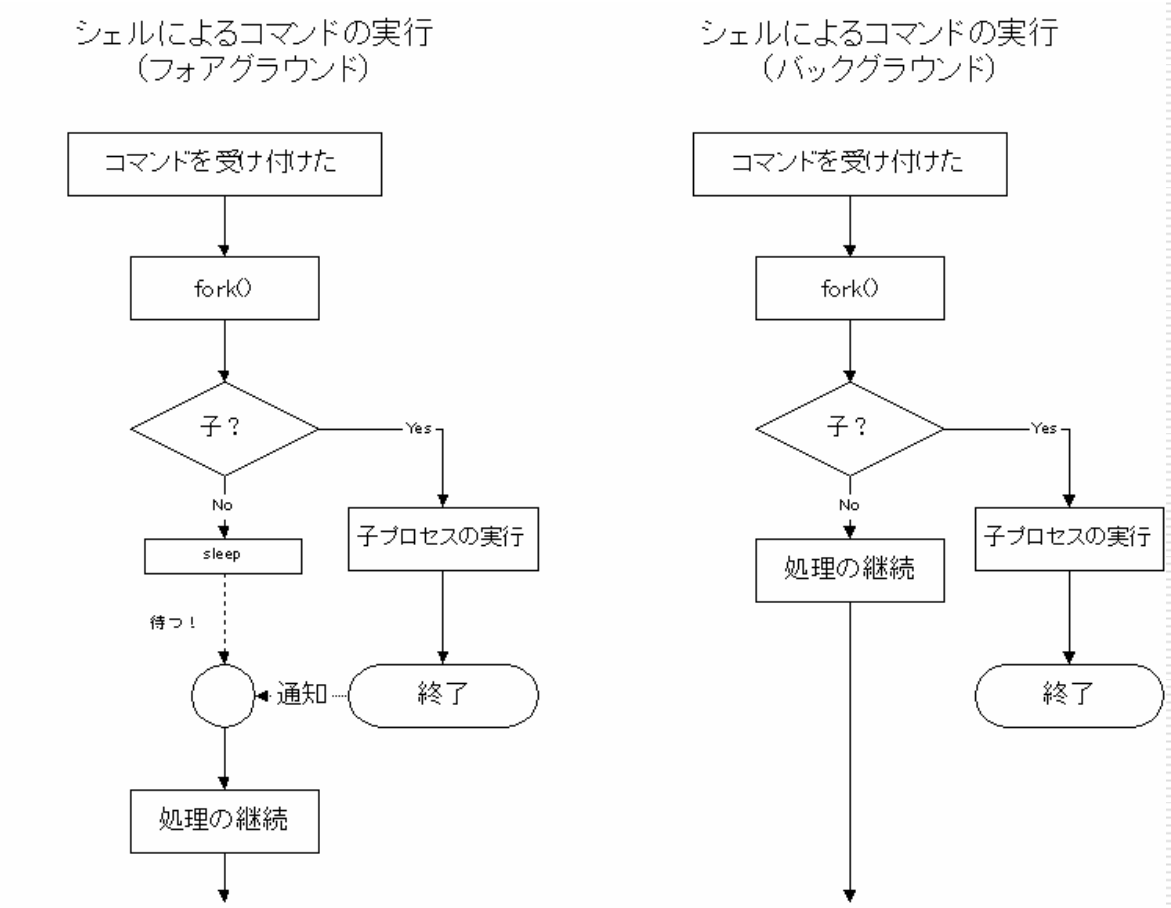
- シェルが、プログラムを実行する。
  - システム関数exec()を「実行」する。
- プログラムを「実行」するとは、どういうことか？
  - プログラムをメモリにロードする。
  - データ領域を確保する。
  - 「命令語」を一行ずつCPUに送る。
  - 「命令語」を実行する。
- 現在実行されているプログラムが、「プロセス」
  - プロセスの状態が、システムで保存されて、OSによって資源が管理される。

関数execl, execv, execleなど

- Exec　= Execute （実行する）
- L　--　LIST,　V　-- Vector（配列）

- execをコールすると、そのプログラム自身がexecで渡した「引数」のプログラムに置き換わる。
  - プログラムが引き継がれる。

シェルによるコマンドの実行



実習課題1（バックグラウンドタスク）

- 時間のかかるプログラムを作成し、そのプログラムをフォアグラウンドで実行させる。
- 同じプログラムをバックグラウンドで実行させる。
- シェルの反応を調べる。

do_forkを調べる

- /usr/src/mmでdo_forkをトレースする。
  - /usr/src/mm/forkexit.c　34行目
- なぜ、mmか？
  - 新たにプロセスが生成された時に、そのプロセスが使用するメモリが割り付けられる。
  - alloc_mem()のコール：57行目
- メモリの管理：教科書P117

メモリ管理の構造体

- mproc構造体
  - /usr/src/mm/mproc.h
  - メンバー: struct mem_map mp_seg[3];

- mem_map構造体
  - /usr/include/minix/type.h

```
struct mem_map {
    vir_clicks   mem_vir;   /* virtual address */
    phys_clicks  mem_phys;/* physical address*/
    vir_clicks   mem_len;  /* length */
}
```

この、virtual address と physical addressとは何か？

教科書： P131 「仮想メモリ」

「ページ」単位の切り替え

- ページ管理
  - メモリを単位する管理として、「ページ」という概念がある。
  - ページイン／ページアウト
  - 実メモリにデータを書き込む際に、アドレスを変換する。
- プロセスが生成（終了）された際に、ページ管理テーブルのエントリーが作成される。
  - これがどこにあるか・・・。

第11回:宿題:「仮想メモリ」

(1)「仮想メモリ」と「物理メモリ」の違いをまとめなさい。

(2)プログラム中に、segmentという概念が扱われている。どのような目的でセグメントが使用されているか、まとめなさい。

ボーナス課題

(3) (1)と(2)で記述した内容を、うまく説明するようなMINIXのソースコードがどこに書かれているか、探して示しなさい。