INE 5412 Sistemas Operacionais I - 2011.2

Laboratório 3 – API – UNIX – Utilizando a API de gerência de processos

Entrega: 06/09/2011 até as 18:00hs

Exercício 3.1 Micro-shell (utilizando chamadas fork(), exec(), exit() e wait())

Descrição:

Implemente um micro-shell capaz de reconhecer e executar comandos padrão do Unix com suas opções e seus argumentos. O micro-shell deve reconhecer ainda pelo menos um comando interno(**exit**) para encerrar sua execução.

Bonus: reconhecer e executar o comando interno **cd** (mudar diretório).

Exemplo de sessão do micro-shell:

```
1    bash $ microshell
2    mshell > ls
3    dir1   dir2   arq1.txt      arq2.txt      arq3.txt
4    mshell > cat arq1.txt
5    este e' o conteudo
6    mshell > exit
7    bash $
```

Exercício 3.2 Temporizador utilizando sinais (utilizando chamadas alarm() e signal())

Descrição:

A chamada de sistema `alarm()` é usada para pedir ao sistema enviar para o processo um sinal especial, chamado `ALRM`, depois de um dado numero de segundos. Considerando que a maioria dos sistemas Unix-like não operam como sistemas de tempo real, seu processo pode receber este sinal depois de um certo tempo da solicitação. Combinando esta chamada de sistema com um tratador de sinal apropriado podemos suportar temporizadores (*timers*) que são importantes para permitir que possamos verificar *timeouts*.

Escreva uma programa que espera por entrada do usuário e termina se depois de um determinado tempo (segundos) nenhuma entrada for recebida. Como funcionalidade básica o programa deve:

- Informar o tempo (seg) como parâmetro do programa.
- No término o programa deve indicar (mensagem) que esta terminando em função de *timeout.*

Exemplos de utilização das chamadas fork(), exec(), exit(), wait() em :

http://www.inf.ufsc.br/~fernando/ine5355/programas/proc

Exemplos de utilização das chamadas signal(), alarm(), kill() em :

http://www.inf.ufsc.br/~fernando/ine5355/programas/sinc