LICENCIATURA EM ENGENHARIA DE SISTEMAS E INFORMÁTICA
LICENCIATURA EM MATEMÁTICA E CIÊNCIAS DA COMPUTAÇÃO

Sistemas Operativos I

Época Normal
1ª Chamada[1]

15 de Janeiro de 2004 Duração: 2h30m

I

1. Compare a segmentação (pura) e a paginação do ponto de vista da alocação de espaço em memória central. Dê uma ideia aproximada dos algoritmos e estruturas de dados usados em cada caso. Que conclusões tira quanto à eficiência?

2. Muitos sistemas operativos actuais fazem o escalonamento de processos usando múltiplas filas. Explique como funcionam e procure demonstrar as suas vantagens face ao ”round robin”. Admita que o computador em causa presta serviço de ficheiros e é por vezes usado para desenvolvimento de algumas aplicações.

II

Pretende-se desenvolver um programa que procurará garantir a permanente execução de um conjunto de programas durante um determinado período de tempo. A invocação do programa deverá ser realizada de acordo com exemplo abaixo. Durante o período de controlo, deverão ser reexecutados todos os programas que entretanto tenham terminado. Findo esse período, todos os programas deverão ser terminados. Apresente o código-fonte do programa de controlo proposto.

```
$ controlador 10 prog1 prog2 ... progN
```

III

Existe num ambiente industrial um conjunto de programas monitores que fornecem valores de sensores do sistema (exemplos: tempA, tempB, presA, veloC, etc). Os valores são todos números inteiros e enviados para o standard output dos programas. Escreva um programa que recebendo como argumentos um número arbitrário destes monitores: 1) os execute concorrentemente; e 2) apresente tanto no seu standard output como num ficheiro de log os valores que vão sendo fornecidos pelos monitores.

Protótipos das chamadas ao sistema relevantes

Processos

- `pid_t fork(void);`
- `void exit(int status);`
- `int execvp( const char *file, char *const argv[]);`
- `pid_t wait(int *status);`
- `pid_t waitpid(pid_t pid, int *status, int flags);`
- `WEXITSTATUS(stat);`
- `int execlp( const char *file, const char *arg, ...);`

Sistema de Ficheiros

- `int open(const char *pathname, int flags, mode_t mode);`
- `int creat(const char *pathname, mode_t mode);`
- `int close(int fd);`
- `int read(int fd, void *buf, size_t count);`
- `int write(int fd, const void *buf, size_t count);`
- `int pipe(int filedes[2]);`
- `int dup(int oldfd);`
- `int dup2(int oldfd, int newfd);`

Sinais

- `void (*signal(int signum, void (*handler)(int)))(int);`
- `int kill(pid_t pid, int signum);`
- `int alarm(int seconds);`
- `int pause(void);`

[1] Cotação — 8+5+7