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

Sistemas Operativos I

Época Normal
1ª Chamada[1]

20 de Janeiro de 2005 Duração: 2h30m

I

Hoje em dia é habitual encontrar em execução num computador várias cópias do mesmo programa: por exemplo, várias janelas ”xterm”cada uma a executar uma ”shell”, vários browsers, servidores web, etc.

1. Acha que a situação descrita acima seria viável em sistemas de memoria real? Porquê? Que problemas surgiriam?

2. Explique em que medida um sistema de memória virtual paginada resolve (ou não) os problemas levantados na alínea anterior.

3. Sabendo que um computador tem normalmente largas dezenas de processos criados, vários deles idênticos, e um só CPU, proponha uma estratégia de escalonamento de processos que considere adequada. Não se esqueça de explicar se tem ou não desafectação forçada, e de justificar as suas opções.

II

O programa **despertador** recebe como argumentos um programa e um inteiro, eg. `despertador prog 10`, permitindo que o programa seja executado ao fim do número de segundos indicado (10 no exemplo dado). No caso de o **despertador** receber um sinal SIGUSR1 antes do prazo indicado, um novo prazo de 5s, a contar da recepção do sinal, deverá ser assumido. Implemente o programa **despertador**.

III

Considere um programa **filtro** que quando invocado com três argumentos: `filtro progA progB ficheiro` processa todo o seu *standard input* em cadeia através dos programas `progA` e `progB` e guarda o resultado em `ficheiro`. O programa **filtro** está preparado para receber o sinal SIGABRT; nessa altura interrompe o processamento escrevendo no ficheiro `"INTERROMPIDO"`, terminando todos os processos filho e a sua própria execução.

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