

Sistemas Operativos I

Segunda chamada¹

26 de Janeiro de 2006

Duração: 2h30m

I

1. Suponha que, ao instalar um sistema operativo num computador portátil para uso escolar, lhe aparece uma “caixa” que apenas diz *Grau de Multiprogramação*. Pode ajustar esse valor entre 1 e 999, ficando a 1 por omissão. Explique os critérios em que se basearia a sua escolha e justifique o valor que escolheu.

2. Suponha que um determinado programa demora a executar 1 minuto no tal computador portátil para uso escolar, quando não há mais nenhum programa em execução. Pretende-se que calcule o tempo médio de resposta se fosse executado simultaneamente com mais 4 processos idênticos, nos casos da estratégia de escalonamento ser FIFO e RR, respectivamente. Escolha uma fatia de tempo adequada ao computador em causa e apresente as contas que fez para chegar aos resultados. Que conclusões tira?

II

Implemente o programa **total** que permite calcular o número total de linhas de diversos ficheiros de texto. O programa em questão recebe o nome dos ficheiros (*pathnames*) através do seu *standard input* e recorre a um programa externo para determinar o número de linhas de cada um deles.

O programa externo - **contarlinhas** - recebe através da sua linha de comando o nome de um ficheiro de texto e produz como output o número linhas (número inteiro em formato binário - 4 bytes). O programa principal, para poder apresentar o resultado final, deverá executar concorrentemente diversos processos auxiliares capturando o seu output.

III

O programa **speakers** recebe como argumento um inteiro (que corresponde a um determinado altifalante a controlar) e está preparado para ser “ligado” e “desligado” através dos sinais SIGUSR1 e SIGUSR2 respectivamente. Inicialmente todos os altifalantes estão desligados.

Pretende-se que codifique o programa **spnet** que recebe como argumento um inteiro n e lança n instâncias do programa **speakers**. O **spnet** recebe no seu *standard input* inteiros de 1 a n separados por ‘\n’. Sempre que o **spnet** recebe um destes inteiros, por exemplo o 5, deverá “desligar” o **speakers** que estiver ligado e “ligar” a 5ª instância de **speakers** que criou. Pretende-se no entanto que a frequência das alterações aos **speakers** não seja superior a 1 por minuto.

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);`

¹Cotação — 8+5+7