

## Sistemas Operativos I

*Época especial<sup>1</sup>*

11 de Outubro de 2006

Duração: 2h00m

### I

1. Explique o que entende por escalonamento com desafectação forçada. Enumere as suas vantagens e desvantagens em relação ao escalonamento cooperativo.
2. Compare em termos de eficiência e protecção a paginação e a multiprogramação com partições de dimensão fixa. Admita que se trata de um sistema que suporta simultaneamente vários utilizadores.
3. Suponha que, perante uma situação em que se verifica uma enorme lentidão num computador e se ouve ininterruptamente actividade de disco, um amigo seu lhe diz que vai comprar um processador mais rápido. Descreva o conselho que lhe daria, justificando os motivos por detrás deste.

### II

Implemente o programa **gongo** que permita imprimir mensagens de alerta. O agendamento das mensagens é especificado através de um ficheiro de texto em que cada linha é composto por o número de minutos que deverão decorrer e uma mensagem.

O programa em questão deverá ficar a repetir o alerta de minuto a minuto até este ser cancelado pelo sinal SIGUSR2. Deverá ainda poder reler o ficheiro de configuração ao receber o sinal SIGHUP e cancelar o alerta associado ao próximo evento previsto ao receber o sinal SIGUSR1.

```
$ gongo ficheiroagenda
```

### III

Desenvolva o programa **fisga** abaixo que deverá lançar dois executáveis interligando reciprocamente os seus standard input e output e limitando a comunicação em qualquer sentido a um valor máximo de bytes especificado na linha de comando.

```
$ fisga progA progB maxBytes
```

#### 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(status);`
- `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);`

<sup>1</sup>Cotação — 8+12