

# Sistemas Operativos

Teste<sup>1</sup>

13 de Junho de 2012

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 Justifique porque, no que diz respeito a algoritmos de substituição de páginas, não é usado normalmente o LRU (*Least Recently Used*), apesar de este se aproximar razoavelmente do óptimo. Descreva um dos algoritmos usados na prática.

3 Descreva os problemas de usar um simples *index block* na estratégia de alocação indexada do espaço em disco. Explique como estes são resolvidos pelo esquema adoptado em UNIX (*inodes*).

## II

Escreva em C um programa que replique para o standard output tudo o que receber do standard input, e que permita inserir a cada segundo uma linha com o conteúdo "PASSOU MAIS UM SEGUNDO". O comportamento de inserção deverá poder ser ligado ou desligado através do envio dos sinais SIGUSR1 e SIGUSR2 respectivamente.

## III

Escreva em C um programa que tem o mesmo comportamento (em termos de criação de processos e execução de programas) que o seguinte comando executado na shell:

```
ls -la | cut -c 20-30 | grep -v echelinho
```

Será valorizada uma solução aplicável a cenários genéricos de encadeamento de comandos.

### Protótipos de algumas funções e chamadas ao sistema relevantes

#### Processos

- `pid_t fork(void);`
- `void exit(int status);`
- `pid_t wait(int *status);`
- `pid_t waitpid(pid_t pid, int *status, int options);`
- `WIFEXITED(status);`
- `WEXITSTATUS(status);`
- `int execlp(const char *file, const char *arg, ...);`
- `int execlpvp(const char *file, char *const argv[]);`
- `int execlpe(const char *file, char *const argv[], char *const envp[]);`

#### Sinais

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

#### 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);`

<sup>1</sup>Cotação — 7+6+7