

Sistemas Operativos I

Primeira chamada¹

15 de Janeiro de 2007

Duração: 2h30m

I

1. É costume dizer-se que *conveniência* e *eficiência* são objectivos dos sistemas operativos. Justifique a afirmação, escolhendo 2 exemplos adequados. Assegure-se que explica convenientemente em que consistem os seus exemplos e que demonstra que os objectivos são de facto atingidos.
2. Num sistema de memória virtual com paginação, a tradução de endereços é normalmente feita recorrendo a tabelas de páginas com múltiplos níveis. Qual o objectivo e que custos tem? Assegure-se que na sua resposta ficou clara a forma como as desvantagens são minimizadas.
3. Explique o funcionamento dos sinais em Unix. Pretende-se uma resposta que descreva sucintamente o que se passa dentro do sistema operativo.

II

Escreva um programa **escalonador** que, quando invocado com n programas como argumentos da sua linha de comando, os escalona em *round robin*. Considere uma fatia de tempo de T segundos. Tenha em conta a terminação dos programas a escalonar.

Utilize primitivas de gestão de processos e sinais do sistema Unix.

III

Pretende-se implementar o programa **mboxsplitter** que deverá permitir separar mensagens de correio electrónico, recebidas através do seu standard input, segundo os seguintes critérios: *i)* as mensagens de mail que contenham spam deverão ser enviadas para o ficheiro `spam.mbox`, *ii)* as mensagens de mail que contenham virus deverão ser enviadas para o ficheiro `virus.mbox` e *iii)* todas as outras deverão ser armazenadas no ficheiro `mail.mbox`.

Para efectuar as operações anteriores dever-se-á recorrer a duas aplicações externas **spamid** e **virusid** que aceitam mensagens de email no seu standard input e as armazenam num ficheiro de texto (especificado na sua linha de comando) ou as copiam para o seu standard output caso satisfaçam ou não o respectivo critério de catalogação.

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);`

¹Cotação — 8+5+7