

Sistemas Operativos*Teste*

25 de Maio de 2021

Duração: 2h

Por favor responda ao grupo I e a cada exercício do grupo II em folhas de teste separadas. Obrigado.**I**

1 Distinga os conceitos de *interrupção* e *desafecção forçada*. Explique os objectivos de cada um destes mecanismos e dê exemplos que considere adequados.

2 Descreva em que consiste o fenómeno conhecido por *thrashing*. Explique como a interacção entre o sistema de memória virtual e um escalonador ingénuo que não tenha em conta o uso da memória, mas apenas a utilização do CPU, pode levar à ocorrência de *thrashing*.

II

Pretende-se criar um banco de dados de registo e pesquisa relativamente aos cidadãos vacinados contra o COVID-19 em Portugal. Assuma que os clientes enviam para o servidor, através de pipes com nome, linhas de texto contendo o número de cartão de cidadão, idade, e código da região; e.g., 07423175 55 2.

1 Implemente o programa servidor, que deve guardar a lista de pacientes vacinados, em diferentes ficheiros, um por região, escrita linha a linha, com o mesmo formato enviado pelos clientes. O servidor deve ser um programa sequencial, mas preparado para a possibilidade de vários clientes se ligarem simultaneamente.

2 Escreva uma função `int vacinados(char* regioao, int idade)`, para ser usada do lado servidor, que devolva o número de pacientes de uma certa idade que foram vacinados numa dada região. Faça uso dos comandos `grep` e `wc`. Por exemplo, `grep " 60 " ficheiro | wc -l` escreve no stdout o número de linhas encontradas no ficheiro correspondentes a pacientes com 60 anos.

3 Escreva uma função `bool vacinado(char* cidadao)`, para ser usada do lado servidor, que permita saber se um certo paciente já foi vacinado. Recorra ao comando `grep cidadao ficheiro`, para pesquisar em cada ficheiro (`grep` termina com exit status 0 caso encontre ocorrências, ou 1 caso contrário). A função deve explorar possível concorrência e retornar o mais rapidamente possível, parando eventuais pesquisas ainda em curso para esse cidadão, que ficaram redundantes.

*Algumas 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 execvp(const char *file, char *const argv[]);`
- `int execve(const char *file, char *const argv[], char *const envp[]);`

Sistema de Ficheiros

- `int open(const char *pathname, int flags, 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);`
- `long lseek(int fd, long offset, int whence);`
- `int access(const char *pathname, int amode);`
- `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);`