

## Sistemas Operativos

Teste

Duração: 1h30m

23 de maio de 2022

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

### I

1 Suponha que tem 6 tarefas independentes entre si para executar num computador *dedicado* à execução deste tipo de tarefas. Sabe-se que cada uma das tarefas envolve cálculos demorados e que os resultados são escritos só no fim, em ficheiros, para serem analisados manualmente. Que algoritmo de escalonamento de CPU seria apropriado a esta situação? Justifique. *Round Robin* *First in First Out*

2 Arquitecturas modernas suportam *huge pages*, páginas com tamanhos muito maiores (e.g., 2MB) do que o clássico.

(a) Descreva que vantagens páginas deste tamanho podem oferecer. Dê um exemplo de uma estrutura de dados apropriada para ser alocada numa região de memória baseada neste tipo de páginas muito grandes. Justifique a sua escolha.

(b) Em sua opinião, estas *huge pages* deveriam ou não ser usadas para mapear em memória ficheiros de *log*, escritos em modo *append* e lidos sequencialmente à medida que o ficheiro cresce? Note que um log poderá crescer lentamente, caso a ocorrência de eventos a registar seja pouco frequente. Justifique a sua resposta.

### II

Assuma que, numa plataforma de suporte a conversas instantâneas entre utilizadores, existe um servidor que guarda as mensagens dos clientes em diferentes ficheiros de texto. O nome do ficheiro coincide com o nome do autor da mensagem, sendo as respectivas mensagens acrescentadas a esse ficheiro, uma mensagem por linha.

1 Escreva a função `int mensagens(char* palavra, char* ficheiro)`, a ser utilizada no servidor, que devolve um inteiro com o número de mensagens em que uma determinada palavra ocorre num determinado ficheiro. Use os comandos `grep` e `wc`, tendo em conta que, por exemplo, o comando `grep palavra ficheiro | wc -l`, invocado na linha de comandos, escreveria no `STDOUT` o número de linhas em que `palavra` ocorre no ficheiro.

2 Escreva a função `int autores_que_usaram(char* palavra, int n, char* autores[n])`, que devolve, para um conjunto de autores, passado num array de `n` elementos, quantos utilizaram uma determinada palavra nas suas mensagens. A função deve desencadear uma procura concorrente, recorrendo à função desenvolvida na pergunta anterior para acesso a ficheiros (não podendo usar chamadas ao sistema relativas a ficheiros).

### 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 fildes[2]);`
- `int dup(int oldfd);`
- `int dup2(int oldfd, int newfd);`