

Sistemas Operativos 1

Frequência¹

17 de Junho de 2009

Duração: 1h30m

I

1 Explique como surge uma interrupção por falta de página (*page fault*) e descreva as acções executadas pelo sistema operativo para a tratar. Use essa descrição para demonstrar por palavras suas a forma como a paginação permite oferecer espaços de endereçamento muito grandes. Deverá ainda ficar claro que isso é conseguido de forma *eficiente*.

2 Suponha que o grau de multiprogramação de um sistema de memória virtual paginado é 4 e que ao monitorizar esse sistema encontrou:

- utilização de CPU: 0,13, utilização de disco: 0,97
- utilização de CPU: 0,87, utilização de disco: 0,03
- utilização de CPU: 0,13, utilização de disco: 0,03

Para cada uma das alíneas anteriores, decida o que fazer em relação ao grau de multiprogramação (manter, aumentar, diminuir) e JUSTIFIQUE.

II

Implemente o programa C equivalente à correcta execução da seguinte script Bash:

```
#!/bin/bash

for ficheiro in *.c; do
  grep FIXME $ficheiro 2> /dev/null > /tmp/fixmes
  contador=`wc -l /tmp/fixmes | cut -c-8`
  if [ "$contador" != "0" ]
  then echo "ficheiro '$ficheiro' necessita de correcções"
  fi
done
```

III

Implemente um programa `repetidor` que deverá executar um conjunto de programas especificados na sua linha de comando. O programa `repetidor` deverá repetir o seu standard input para os programas especificados.

```
$ repetidor prog1 prog2 ...
```

IV

Implemente um programa `correcto` que deverá inspecionar regularmente o conteúdo de um ficheiro de *logs* especificado na sua linha de comando. A frequência de é igualmente especificada como argumento da linha de comando. O programa terminará a sua execução quando encontrar uma nova ocorrência da expressão "error".

```
$ monitor /var/log/messages 10
```

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

¹Cotação — 8+7+5

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);
-