

Sistemas Operativos

Frequência¹

19 de Junho de 2013

Duração: 2h

I

1 Suponha que o seu computador pessoal está cada vez mais lento e que pode escolher entre: a) desfragmentar o disco; b) duplicar a capacidade da memória central; ou c) substituir o leitor de dvd por um disco de estado sólido (SSD) de capacidade “reduzida” (128 GB). Ignorando os custos das opções b) e c), em que circunstâncias optaria por cada uma dessas alternativas? Não se esqueça de dar detalhes (e.g. usaria o SSD para... o disco está fragmentado porque...) e descrever os benefícios de cada uma das alternativas.

2 Imagine que se usou o algoritmo seguinte para a marcação no Blackboard das apresentações dos trabalhos práticos de diversos grupos de 2 alunos: um dos elementos do grupo procura um “slot” livre, inscreve o seu nome e depois comunica ao seu colega o número do slot escolhido para ele também proceder à inscrição nesse slot. Infelizmente verificaram-se dois casos em que um dos elementos do grupo se inscreveu mas quando o colega tentou já encontrou o slot cheio com um elemento do outro grupo. Os líderes dos dois grupos afectados afirmam que se inscreveram em slots livres. Consegue explicar o que se terá passado? Consegue propor uma solução para apresentar aos responsáveis pelo Blackboard? Que problemas terá essa solução? Pode usar um mecanismo de sincronização à sua escolha (incluindo semáforos).

II

Implemente um programa `krun` que receba como argumento um comando (e os seus argumentos) e assegure que há sempre 5 processos a correr esse comando. Por exemplo, “`krun xterm -ah`” deverá manter em execução 5 processos `xterm` que arrancam com o argumento `-ah`. Um processo que termine a sua execução deve ser substituído por outro. Mediante o envio de `SIGUSR1` e `SIGUSR2` ao processo do `krun`, deverá ser possível incrementar e decrementar o número de processos a manter em execução. Use primitivas de processos e sinais.

III

Pretende-se fazer a locução de documentários legendados por ficheiros no formato “`segs\n frase\n`” (ver exemplo abaixo). Implemente um programa `locutor` que leia o ficheiro de legendas a partir do seu standard input e que, respeitando os tempos indicados, envie cada frase para um conjunto de processos que sintetizarão a voz de locução. Cada processo de síntese de voz recebe as frases linha a linha também a partir dos respectivos standard input. O programa `locutor` deverá permitir a locução simultânea em vários idiomas sendo-lhe passada como argumento a respectiva lista de comandos de síntese de voz.

0

A RUM apresenta

10

As JOIN 2013

...

Protótipos de algumas funções e chamadas ao sistema relevantes

Processos

- `pid_t fork(void);`
- `void exit(int status);`
- `pid_t wait(int *status);`
- `WIFEXITED(status);`
- `WEXITSTATUS(status);`
- `int execlp(const char *file, const char *arg, ...);`
- `int execvp(const char *file, char *const argv[]);`

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 e pipes

- `int open(const char *path, int flags, mode_t mode);`
/* `O_RDONLY`, `O_WRONLY`, `O_RDWR`, `O_CREAT`, `O_TRUNC`, `O_APPEND` */
- `int close(int fd);`
- `int read(int fd, void *buf, size_t count);`
- `int write(int fd, const void *buf, size_t count);`
- `int access(const char *path, int amode);` /* `F_OK`, `R_OK`, `W_OK`, `X_OK` */
- `off_t lseek(int fd, off_t offset, int whence);` /* `SEEK_SET`, `SEEK_CUR`, `SEEK_END` */
- `int unlink(const char *file);`
- `int mkfifo(const char *path, mode_t mode);`
- `int dup2(int fd1, int fd2);`
- `int pipe(int fd[2]);`

¹Cotação — 4+4+6+6