

Sistemas Operativos

Teste

Duração: 2h

8 de Junho de 2018

Por favor responda a cada um dos 3 grupos em folhas de teste separadas. Obrigado.

I

- 1 Admita que um grande conjunto de dados que tem de ser processado em cadeia. Demonstre que neste caso a comunicação entre processos por *pipes anónimos* é muito mais eficiente e conveniente do que a utilização de ficheiros, sendo provável que também contribua para a redução de tempos de resposta.
- 2 Suponha que ao instalar um sistema operativo num computador (eg, linux, a ser usado nas aulas de SO) lhe aparece uma "caixa" que pede para escolher o Grau de Multiprogramação. Por omissão fica a 1 mas pode ajustar esse valor entre 1 e 999. Explique os critérios em que basearia a sua escolha e justifique o valor que escolheu.
- 3 Se um determinado programa demora a executar 5 minutos quando não há mais nada em execução num computador, qual o tempo médio de execução se forem executados simultaneamente 4 processos idênticos, no caso do escalonamento ser FIFO ou RR, respectivamente. Escolha uma fatia de tempo adequada ao computador em causa e apresente as contas que fez para chegar aos resultados. Que conclusões tira?

II

Considere um programa 'controlador' que deverá executar repetidamente um conjunto de programas especificados na sua linha de comando, até um destes terminar com código de saída de 0. Os programas deverão ser todos executados concorrentemente, em tentativas sucessivas. Sempre que numa tentativa nenhum dos processos terminar com código de saída de 0, a próxima tentativa tem como limite de tempo (em segundos) a soma dos códigos de saída. Findo esse limite de tempo, todos os processos, incluindo o controlador, devem ser terminados. A primeira tentativa não tem limite de tempo. Implemente o programa controlador recorrendo às primitivas de processos e sinais estudadas.

III

Escreva um programa, invocado da forma `controlador p c`, para processamento de linhas de texto produzidas concorrentemente por p processos, instâncias de um programa produtor, por c processos, instâncias de um programa consumidor, ambos invocados sem argumentos. Assuma que cada linha produzida começa por um dígito d , e faça com que o resto da linha seja processada pelo consumidor d , quando $0 \leq d < c$, ou descartada caso contrário.

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);`