MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA
LICENCIATURA EM CIÊNCIAS DA COMPUTAÇÃO

Sistemas Operativos

Exame de Recurso

16 de Junho de 2021 Duração: 2h

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

I

1 Relativamente aos estados de um processo, explique o que é um processo "bloqueado" e a importância da existência deste estado. Dê exemplos.

2 Justifique porque, no que diz respeito a algoritmos de substituição de páginas, não é usado normalmente o *Least Recently Used*, apesar de este se aproximar razoavelmente do óptimo. Descreva um dos algoritmos usados na prática.

II

Implemente um programa `memoria` que executa 10 vezes, sequencialmente, um programa (com eventuais argumentos), apresentando no final a memória virtual usada (valor mínimo, médio e máximo das 10 execuções).

```
$./memoria programa arg1 .. argN
memória: 257891 276878 294632
```

Considere que a memória virtual (em KB) usada por um processo pode ser obtida através do standard output resultante da execução de (com `[pid]` substituído pelo pid do processo apropriado):

```
$ grep VmPeak /proc/[pid]/memstats | cut -d" " -f4
```

III

Implemente um programa que corre em rondas sequenciais. Em cada ronda deverão ser executadas 100 instâncias concorrentes do comando `cmd` (sem argumentos). Cada ronda deve terminar no máximo após 20 segundos, devendo eventuais instâncias do comando ainda a executar serem forçadas a terminar. Ao fim de cada ronda, o programa deverá escrever o número de vezes que o comando executou completamente nessa ronda (sem ser interrompido). Ao ser interrompido com SIGINT, o programa deverá escrever o número de rondas finalizadas, abortar a ronda em curso e terminar.

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);