

**Sistemas Operativos I***Época Especial*<sup>1</sup>

15 de Setembro de 2007

Duração: 2h30m

**I**

Considere o programa **piobench** descrito no grupo II.

- 1 Descreva dois dos principais problemas (à sua escolha) com que se depararia se necessitasse de executar o programa **piobench** num sistema de memória real, e explique em que medida a existência de paginação os permitiria ultrapassar. Deverá justificar a sua resposta com base em detalhes de implementação.
- 2 Descreva o que entende por *swapping* e explique porque é relevante mesmo num contexto em que existe memória virtual implementada com paginação.
- 3 Sabendo que o programa **piobench** vai lançar a execução de vários processos, explique em que medida os resultados apresentados poderão ser influenciados pela estratégia de escalonamento de processos. Proponha então uma estratégia que lhe permita ter confiança nos resultados obtidos.

**II**

Implemente um programa **piobench** cujo propósito é o de determinar o tempo médio de envio e recepção – ida e volta – de uma linha de texto através de pipes entre dois processos. Considere que o programa recebe como argumento da sua linha de comando o tempo de amostragem e o tamanho da linha de texto a enviar. Enquanto não se esgotar esse tempo deverá permanecer em ciclo a enviar/receber uma linha de texto, para no final apresentar os resultados no *standard output*. Utilize o programa externo `cat` para o processo repetidor da informação.

**III**

Pretende-se implementar um programa **lancador** que deverá executar um conjunto de programas externos listados num ficheiro de texto especificado na sua linha de comando. Cada linha do ficheiro identifica o programa a executar e se este depende ou não da conclusão dos programas que o precedem. Se o programa **lancador** receber o sinal SIGUSR1 todos os processos em execução deverão ser forçosamente terminados.

*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[]);`

**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);`

<sup>1</sup>Cotação — 7+6+7