Nome (maiúsculas)**:**

Número de aluno:

Instituto Superior de Engenharia de Coimbra	**Sistemas Operativos** 2013/14 DEIS – Engenharia Informática	Prova escrita Parte teórica 12/7/2014

Todas as perguntas valem o mesmo. Respostas erradas descontam 1/3. Respostas múltiplas são respostas erradas.
<u>As repostas são dadas nestas folhas. Preencha já o cabeçalho com o nome.</u>

1 – No contexto dos sistemas estudados, quando o sistema é iniciado ("durante o arranque"), são desencadeadas várias tarefas, as quais incluem necessariamente a interacção com periféricos (por exemplo, ler configurações do disco, apresentar menus de arranque no ecrã, etc.). Como é feita essa interacção com periféricos?

- ☐ É feita com recurso a rotinas existentes na BIOS.
- ☐ É totalmente feita através de código existente no *bootloader*.
- ☐ É feita com recurso a rotinas existentes no sistema operativo (por exemplo, *device drivers*).
- ☐ É totalmente feita através de código existente no MBR.

2 – Onde fica a informação acerca de qual é a partição de arranque?

- ☐ Na memória não volátil (vulgo "BIOS").
- ☐ No menu que é apresentado ao utilizador para escolher qual o sistema que deseja iniciar.
- ☐ No sector MBR.
- ☐ No sector *boot*.

3 – O ficheiro /etc/sudoers tem uma linha que que contém "matt groening=NOPASSWD: /sbin/*". Isto significa que:

- ☐ Os utilizadores matt e groening não precisam de *password* para entrar.
- ☐ Os utilizadores matt e groening podem usar os comandos do *root* em /sbin desde que estes não exijam *password*.
- ☐ O utilizador matt pode executar os comandos em /sbin como root na máquina groening e nem sequer lhe será pedida a *password*.
- ☐ O utilizador matt pode usar os comandos em /sbin como se fosse o utilizador groening desde que este último não tenha *password* definida.

4 – O bit setuid serve para:

- ☐ Permitir/não permitir a execução de programas.
- ☐ Permitir/não permitir aos utilizadores mudarem o seu ID.
- ☐ Permitir correr programas com um ID diferente do utilizador que o executa.
- ☐ Permitir a um processo modificar o seu PID.

5 – O processo A invocou a função biblioteca C *malloc* para alocar memória. Neste instante a função está a devolver com resultado NULL porque não há memória para satisfazer pedido. Por isso, nesse instante, o processo:

- Passou para o estado “executável”.
- Passou para o estado “bloqueado” (até voltar a haver memória).
- Termina.
- Mantém-se no estado “execução”.

6 – No contexto de um sistema como os usados durante as aulas (sistemas multiprogramados, com um processador, e memória virtual), um processo A acaba de receber um sinal enviado pelo processo B. O processo B está neste momento:

- Necessariamente no estado “executável”, até que o processo A termine o atendimento do sinal ou termine.
- No estado “bloqueado”, à espera do processo A.
- Num estado qualquer: qualquer um é possível pois são processos independentes.
- Qualquer um menos execução, inclusivamente até pode ter já terminado.

7 – Numa arquitectura *micro-kernel* / cliente-servidor, qual dos seguintes componentes do sistema pode ser colocado em espaço-utilizador?

- Apenas as aplicações.
- O escalonador de processos.
- O sistema de ficheiros.
- O gestor de dispositivo (“device driver”) do disco rígido.

8 – Qual das seguintes afirmações é correcta?

- Os sistemas preemptivos surgiram para aproveitar os tempos em que um processo se encontra em operações de entrada/saída.
- Os sistemas multiprogramados surgiram para garantir que um processo não fica a ocupar o processador indefinidamente.
- Os sistemas preemptivos são muito bons para servidores, mas não para sistemas interactivos (usados de forma interactiva pelos utilizadores).
- Os sistemas de processamento em lote não conseguem aproveitar bem as operações de entrada/saída para executar outras tarefas.

9 – No contexto dos algoritmos estudados, qual das seguintes afirmações é correcta?

- Os algoritmos mais justos introduzem o problema de *starvation*.
- O algoritmo *round robin* é genericamente bom, mas não é necessariamente o melhor em todos os aspectos.
- Os algoritmos SPN e SRT são mais adequados a ambientes de controlo.
- No algoritmo HRRN resolve o problema de *starvation* e, sendo *preemptivo*, é também muito estável.

Nome (maiúsculas)**:**

Número de aluno:

10 – Assumindo que não há erros de execução, qual poderá ser o resultado da execução do código à direita?

- ☐ Apresenta no ecrã e no ficheiro temp.txt a lista de processos a correr.
- ☐ Apresenta no ecrã a listagem dos processos a correr e não faz mais nada.
- ☐ Apresenta duas vezes no ecrã a listagem dos processos a correr.
- ☐ Coloca no ficheiro temp.txt a listagem dos processos a correr e não faz mais nada.

```
int main() {
   if (fork!=0) {
      close(1);
      open("temp.txt", O_WRONLY | O_CREAT);
   }
   execlp("ps", "ps", null);
   return 0;
}
```

11 – Observe o código do programa "teste" à direita. Assumindo que não há erros nas chamas às funções sistema, o que aparece ao se executar o programa?

- ☐ Nada.
- ☐ "valor 1".
- ☐ "valor 1" oito vezes.
- ☐ "valor 1" "valor 2" …. "valor 7" e "valor 8"

```
int conta = 0;
int main() {
   int i;
   for (i=0; i < 3; i++) {
      fork();
      conta++;
      execl("teste", "teste", NULL);
   }
   printf("valor %d", conta);
   return 0;
}
```

12 – Considere um sistema a correr num sistema cujo *hardware* tem as seguintes características: i) capacidade de endereçamento virtual; ii) um mecanismo de interrupções habitual; iii) um só modo de execução. Neste sistema:

- ☐ Não é possível implementar espaços de memória diferentes para cada processo.
- ☐ Não é possível garantir a estabilidade (funcionamento correcto) do sistema.
- ☐ Os periféricos não conseguem notificar o sistema acerca de acontecimentos (erros, etc.).
- ☐ O sistema não tem limitação nenhuma face aos sistemas habituais.

13 – Considere o programa à direita. Assumindo que não há erros de execução, o que aparece no ecrã?

- ☐ Duas vezes o mesmo número.
- ☐ Dois números diferentes.
- ☐ Três números, dois deles iguais.
- ☐ Três vezes o mesmo número.

```
int main() {
   printf("%d\n", getpid());
   fork();
   printf("%d\n", getpid());
   return 0;
}
```

14 – Num determinado sistema com o algoritmo de escalonamento SPN foram lançados os processos A,B,C,D nos instantes 0,1,2,3, respectivamente. Estes processos têm a seguinte duração, respectivamente: 3,1,3,2. O *turnaround* médio foi:

- ☐ 6 / 4.
- ☐ 11 / 4.
- ☐ 15 / 4.
- ☐ 16 / 4.

15 – No contexto de sistemas de endereçamento virtual,

- ☐ O endereçamento segmentado é mais eficiente e simples de gerir do que o endereçamento paginado.
- ☐ Só se consegue isolar os espaços de endereçamento de processos uns dos outros com endereçamento segmentado (e não com endereçamento paginado): processos diferentes terão segmentos diferentes.
- ☐ Só é viável (prático) implementar memória virtual à custa do disco em sistemas de endereçamento paginado.
- ☐ As tabelas de páginas são normalmente muito menores que as tabelas de segmentos.

16 – Num sistema com memória paginada encontra-se a correr o processo A cuja tabela de páginas é a que se encontra à direita. O sistema é de 32 bits, 12 dos quais usados para o deslocamento (ou seja, páginas de 4096 bytes). Qual o endereço real manipulado quando é executada uma instrução que pretende guardar o valor 123 no endereço virtual 4098?

Índice	P	Prot	Base	R	M
0	1	RWX	32768	0	0
1	1	RWX	8192	0	1
2	1	RWX	16384	1	0
3	0	RWX	0	1	1

- ☐ Endereço 16386 (16384 + 2).
- ☐ Endereço 8194 (8192 + 2).
- ☐ Endereço 12290 (8192 + 4098).
- ☐ Nenhum porque 4098 é maior que o tamanho da página.