Universidade de Aveiro

Departamento de Electrónica, Telecomunicações e Informática

Sistemas de Operação

Exame NM (Ano Letivo de 2016/17) 19 de Janeiro de 2017

Nome: _____ NMec: _____

NOTA: Numa questão em que se peça uma justificação e ela não seja dada, a resposta não será considerada.

. . . . . .

1. Considere o excerto de código seguinte, parte de uma implementação do problema do jantar dos filósofos. Cada filósofo é implementado por uma *thread*, que, após inicialização, executa a função `philosopher`, e é identificado pelo parâmetro `id`, que varia entre 0 e N-1.

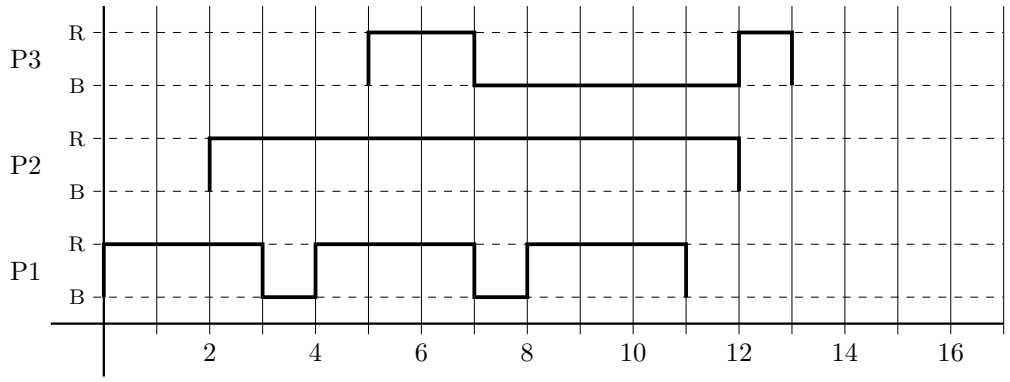
```
1  sem_t fork[N];
2
3  #define left(i) i
4  #define right(i) ((i+1)\%N)
5  void philosopher(int id)
6  {
7      while (true)
8      {
9          think();
10         down(fork[left(id)]);
11         down(fork[right(id)]);
12         eat();
13         up(fork[left(id)]);
14         up(fork[right(id)]);
15     }
16 }
```

   (a) A possibilidade de ocorrência de *deadlock* pressupõe a satisfação em simultâneo de 4 condições. Quais são?

   (b) A implementação do jantar dos filósofos apresentada pode conduzir a *deadlock*. Mostre que as 4 condições anteriores são satisfeitas.

   (c) Altere o código dado de modo a evitar a ocorrência de dealock. Explique a sua solução, indicando a(s) condição(ões) que negou.

2. O gráfico seguinte representa o estado da execução de 3 processos independentes entre si (mesmo em termos de I/O), P1, P2 e P3, assumindo que correm em processadores (virtuais) distintos. R e B indicam, respetivamente, que o processo está no estado `RUN` (a usar o processador) ou no estado `BLOCKED` (bloqueado à espera de um evento).

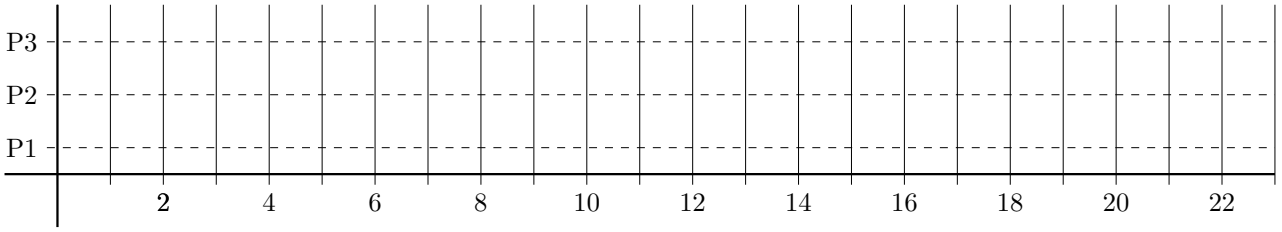


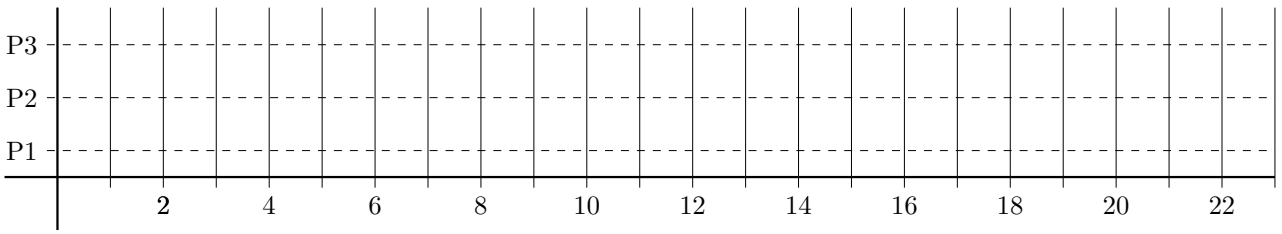
(a) Distinga processos CPU-intensivos e processos I/O-intensivos. Em que categoria coloca os processos P1, P2 e P3? Justifique a sua resposta.

(b) Compare as políticas de escalonamento do processador designadas por FCFS (First Come First Served) e Round Robin.

(c) Distinga multiprocessamento e multiprogramação.

(d) Considere que os 3 processos representados acima correm num ambiente monoprocessador. Usando os gráficos abaixo, trace os diagramas temporais de escalonamento do processador pelos processos P1, P2 e P3, considerando as políticas de escalonamento FCFS e Round Robin, esta com um *time quantum* (*time slot* atribuído a cada processo) de 2.





3. Considere o programa apresentado a seguir, onde `delay()` é uma função que gera um atraso com tempo aleatório em *busy waiting* (ou seja, não bloqueante).

```
1  int main(void)
2  {
3      printf("msg 0\n");
4      int pid = fork();
5      switch (pid)
6      {
7          case 0:
8              delay();
9              printf("msg 1\n");
10             printf("msg 2\n");
11             break;
12         default:
13             delay();
14             printf("msg 3\n");
15             wait(NULL);
16             printf("msg 4\n");
17     }
18     return 0;
19 }
```

(a) Assumindo que o `fork` não falha, que linhas do programa anterior são executadas no processo pai e no processo filho? Justifique sucinta e adequadamente a sua resposta.

(b) Considerando que a execução de um `printf` é atómica, além da saída

```
msg 0
msg 1
msg 2
msg 3
msg 4
```

apresente outras possíveis saídas (em termos de *standard output*) que podem resultar da execução do programa anterior. Justifique sucinta e adequadamente a sua resposta.

(c) O escalonador de processador de baixo nível típico possui 3 estados, normalmente designados de `RUN`, `READY_TO_RUN` e `BLOCKED`. Trace o diagrama de estados para um escalonador de baixo nível, considerando os estados anteriores. Para cada transição considerada, explique o seu papel e quando é que ocorre.

(d) Considerando que a execução do programa resulta na saída apresentada na alínea (b) e que os `printf` nunca bloqueam o processo, o processo pai pode passar pelo estado `BLOCKED`? Justifique sucinta e adequadamente a sua resposta.

4. Considere um sistema de memória virtual paginada, onde a cada processo é atribuído um máximo de 10 páginas e em que a memória principal do sistema tem 5 *frames*. Considere ainda que um processo (único) executou a seguinte sequência de referências, em termos de páginas de memória acedidas: 1, 2, 3, 4, 5, 3, 4, 9, 6, 7, 1, 7, 8, 1, 7, 8, 9, 5, 4, 5, 2.

   (a) A tabela seguinte representa, ao longo do tempo, as páginas residentes nas *frames* de memória. Complete-a considerando que o algoritmo de substituição de páginas utilizado é o FIFO. Preencha apenas as células da tabela quando há mudança de página.

	1	2	3	4	5	3	4	9	6	7	1	7	8	1	7	8	9	5	4	5	2
F5					5								8								
F4				4							1										
F3			3							7											2
F2		2							6										4		
F1	1							9										5			

   (b) A tabela seguinte representa, ao longo do tempo, as páginas residentes nas *frames* de memória. Complete-a considerando que o algoritmo de substituição de páginas utilizado é o LRU (Least Recently Used). Preencha apenas as células da tabela quando há mudança de página.

	1	2	3	4	5	3	4	9	6	7	1	7	8	1	7	8	9	5	4	5	2
F5					5					7											2
F4				4									8								
F3			3								1								4		
F2		2							6									5			
F1	1							9													

   (c) O algoritmo LRU tem um custo de implementação elevado e é pouco eficiente. Uma aproximação menos exigente e relativamente eficiente é o algoritmo NRU (Not Recently Used). Descreva o princípio de funcionamento deste algoritmo.

. . . . . .

5. Considere o sistema de ficheiros `sofsxx`, semelhante aos `sofs15` e `sofs16`, com blocos de tamanho 512 bytes (2^9) e *clusters* de 4 blocos.

   (a) Sabendo que as referências aos clusters têm 32 bits e desprezando os blocos usados pelo superbloco e pela tabela de nós-i (*inodes*), calcule o tamanho máximo em bytes que um disco formatável em `sofsxx` pode ter? Apresente os cálculos necessários para justificar a sua resposta.

   (b) Considerando que um nó-i do `sofsxx` possui 5 referências diretas, 1 indireta, 1 duplamente indireta e 1 triplamente indireta, calcule o tamanho máximo em bytes que um ficheiro pode ter? Apresente os cálculos necessários para justificar a sua resposta.

   (c) O `sofsxx` suporta *hard links* e *soft links* (atalhos). Explique a diferença entre ambos.