

Número:

Nome:

LEIC/LETI - 2015/16 - Repescagem do 2º Teste de Sistemas Operativos

27 de Janeiro de 2016

Responda no enunciado, apenas no espaço fornecido. Identifique todas as folhas.

Duração: 1h

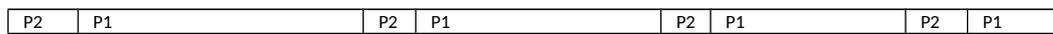
Grupo I [10 Val]

1. [1 v] “A comutação de processos ou tarefas ocorre exclusivamente quando um processo ou tarefa em execução chamam uma chamada sistema.”
A afirmação é verdadeira ou falsa? Justifique.
-

-
2. [1,5 v] Assuma um escalonador de tempo partilhado *round-robin*. Ao instalar o sistema operativo pode escolher entre *time slices* de 10 ms e 50 ms. Apresente uma vantagem de cada opção.
-

3. Considere agora um sistema Linux com 1 CPU onde, num dado momento, existem 2 processos ativos.

Ao longo de um período de tempo, monitorizou-se qual o processo em execução e obteve-se a seguinte informação:



tempo

Ambos os processos têm prioridades base iguais. Assuma também que, no início do período acima, ambos os processos começam com o mesmo valor de quantum por usar.

- a. [1 v] Apresente possível uma razão para P1 ter maiores fatias de execução que P2 no exemplo acima.

- b. [1,5 v] No final da execução acima, qual será o processo mais prioritário? Justifique explicitando a fórmula como o escalonador do Linux estudado nas aulas teóricas calcula as prioridades.

- c. [1 v] Assuma que, mais tarde, são criados mais 20 processos no sistema. A duração de uma época do escalonador Linux aumentará, diminuirá, ou manter-se-á constante? Justifique.

- d. Considere as comutações de processo, entre P1 e P2.

- i) [1 v] Dê 3 exemplos de elementos do contexto software que mudam quando se comuta de processo.

- ii) [1 v] Dê 3 exemplos de elementos do contexto hardware que mudam quando se comuta de processo.

4. Num programa concorrente que usa variáveis de condição para sincronização entre tarefas (usando a API *pthread*), considere o seguinte excerto:

```
1 pthread_mutex_lock(&mutex);  
  ...  
2 pthread_cond_signal(&condicao);  
  ...  
3 pthread_mutex_unlock(&mutex);
```

a. [1 v] Indique, em pseudo-código, o que faz a linha 2.

b. [1 v] Imediatamente após o retorno de `pthread_cond_signal`, pode a tarefa chamadora perder o processador para uma tarefa que estava bloqueada na fila de espera da mesma variável de condição?

Grupo II [10 Val]

Considere a seguinte instrução de um programa que abre um ficheiro mantido num sistema de ficheiros EXT, numa máquina Linux:

```
int f = open("/home/maria/readme.txt", O_RDONLY);
```

1. [1,5v] Para executar a função, será necessário ler da região de blocos do volume em disco (ou da cache de blocos)? Se sim, indique qual/ quais são esses blocos.

2. [2v] Assumindo que i) o processo que executou a instrução acima não tinha aberto nenhum ficheiro antes (ou seja, tem apenas o *stdin*, *stdout* e *stderr* abertos) e ii) a função retornou com sucesso, apresente numa figura o conteúdo de: tabela de ficheiros abertos do processo e tabela de ficheiros abertos global. Complete a sua figura com toda a informação que este enunciado permite determinar.

3. [2v] Assuma que a instrução seguinte é `fork()`. Apresente o conteúdo das tabelas de ficheiros abertos e tabela de ficheiros global depois dessa instrução completar.

4. Assuma que o i-node do ficheiro em causa tinha, na sua tabela de índices de blocos, o seguinte conteúdo:

Referências diretas												Ref. Indiretas		
45	46	47	48	49	50	51	52	53	54	55	56	178	0	0

Considere também que:

- no volume onde o ficheiro se encontra, os blocos são de 4 kBytes
- a dimensão do ficheiro é 81 kBytes

- a. [1,5v] Em qual bloco do disco se encontra o último byte deste ficheiro? Responda indicando o índice desse bloco. Caso o enunciado não lhe permita responder, justifique.

- b. Apesar deste ficheiro ter dimensão de 81 kBytes, na prática ele ocupará um maior espaço em disco.

- i) [1,5v] Indique quanta fragmentação interna (em bytes) existe nos blocos de dados deste ficheiro. Justifique.

- ii) [1,5v] Indique que meta-dados associados a este ficheiro são mantidos em disco. Justifique.
