Número: Nome:

LEIC/LETI – 2019/20 – 2º Teste - 21 de janeiro de 2020

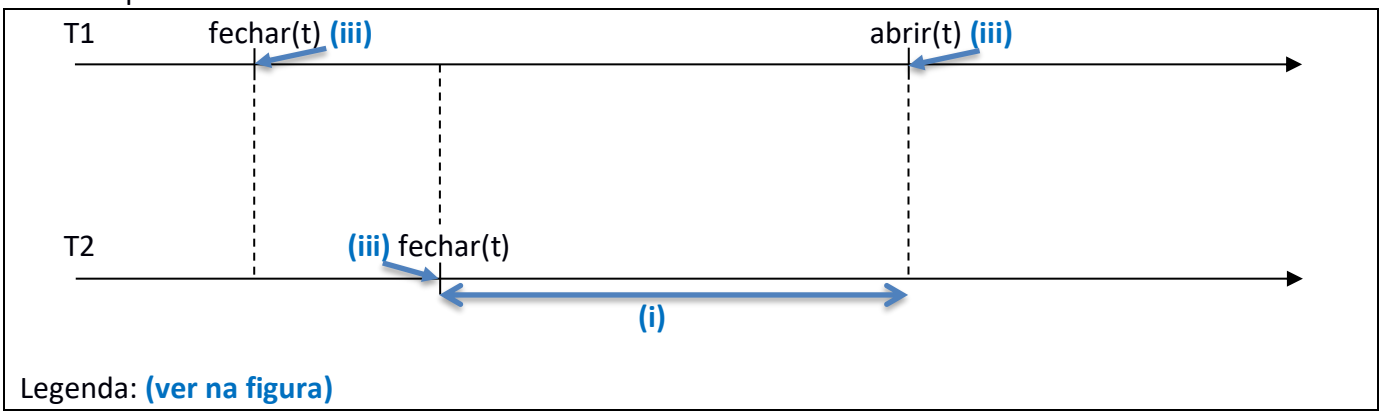
- **Duração: 1h30m**
- **Responda no enunciado, apenas no espaço fornecido. Identifique todas as folhas.**
- **Nas perguntas de escolha múltipla, uma resposta errada desconta 1/2 ou 1/3 da cotação (consoante haja 3 ou 4 opções de escolha, respetivamente).**

Grupo I [8 Val]

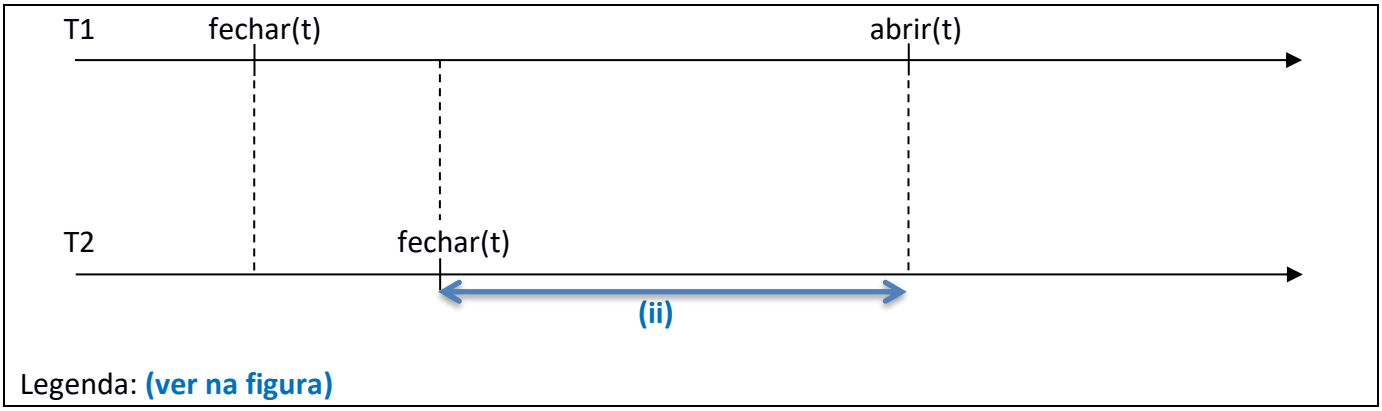
1. Um processo tem duas tarefas, *T1* e *T2*, que trabalham sobre dados na memória partilhada do processo. Para tal, sincronizam o acesso às secções críticas através de um trinco, *t*.
   Nas suas respostas abaixo, considere que cada tarefa se executa num *core* distinto; e que, enquanto *T1* ou *T2* estão em execução, não perdem o processador para outras tarefas/processos.

   Nas duas alíneas seguintes, responda **marcando sobre as barras do tempo** e junte uma **legenda** que indique claramente o significado de cada marcação.

   a) [1v] Comece por assumir que o trinco *t* é um trinco **suportado pelo núcleo.** No exemplo de execução ilustrado abaixo, assinale claramente: i) os períodos em que alguma tarefa esteve bloqueada; ii) os períodos em que alguma tarefa esteve em espera ativa; iii) os momentos em que ocorreram chamadas sistema.



   b) [1v] Responda de novo sobre o exemplo abaixo, mas agora assumindo que o trinco *t* é um **trinco *hardware*, implementado numa biblioteca em modo utilizador** recorrendo a uma instrução atómica (como BTS ou XCHG).



2. Considere o escalonamento baseado em épocas que foi usado em versões anteriores do Linux, tal como estudado nas aulas.

a) Neste algoritmo, a prioridade de um processo é calculada por:
Prio = prio_base + quantum_por_usar_nesta_época – nice

i) [1v] Com base na fórmula acima, indique se este algoritmo privilegia processos ES-intensivos (*IO-bound*) ou CPU-intensivos e justifique sucintamente.

Privilegia processos: **ES-intensivos**

Justificação: **Processos ES-intensivos usam pouco tempo de execução pois bloqueiam-se frequentemente, logo o seu quantum por usar tenderá a ser superior aos processos CPU-intensivos. Consequentemente, a fórmula acima determina que os processos ES-intensivos tendem a ter um valor de prioridade superior.**

ii) [0,8v] No caso de um processo que corre um programa como o referido na pergunta 1, optar entre um trinco com espera bloqueante e um trinco com espera ativa tem algum impacto na prioridade calculados pelo escalonador?

A) Com trinco de espera bloqueante, a prioridade das tarefas em espera tenderá a ser maior.
B) Com trinco de espera bloqueante, a prioridade das tarefas em espera tenderá a ser menor.
C) O tipo de espera (bloqueante ou ativa) não tem qualquer impacto nos cálculos do escalonador.

A

b) No início de uma época, o quantum disponível a um processo nessa época é dado por:
quantum = quantum_base + quantum_por_utilizar_epoca_anterior / 2

i) [1v] Descreva uma situação em que, quando a época termina, um dado processo não usou todo o quantum que tinha disponível.

Um processo fez chamada sistema que o bloqueou durante longo período. Se, durante esse período de bloqueio, a época terminar, o processo bloqueado transita para a próxima época sem ter usado todo o seu quantum.

iii) [0,8v] Qual a razão da divisão por 2 na fórmula acima?

A) Reflete o facto de a máquina ter 2 CPUs.
B) Permite favorecer os processos que na época anterior usaram mais CPU.
C) Permite suportar preempção.
D) Permite dar maior importância às utilizações recentes do processador por parte deste processo, esquecendo progressivamente as utilizações mais antigas.

D

3. Considere o seguinte excerto de programa que, iterativamente, calcula o valor de pi com um número crescente de casas decimais.
Assuma que a função *calculaMaisUmaCasaDecimal* apenas executa instruções aritméticas (em modo utilizador, sem realizar chamadas sistema).

```
float pi = 3;

void terminaComputacao(int s) {
  printf("Ok, vou parar. Resultado: %f\n", pi);
  exit(0);
}

float calculaMaisUmaCasaDecimal(float x) {...}

int main() {
  signal(SIGINT, terminaComputacao);
  while (1)
    pi = calculaMaisUmaCasaDecimal(pi);
  return 1; //Nunca deve chegar aqui
}
```

a) [0,8v] Que estrutura mantida pelo núcleo é alterada pela chamada *signal*?
   A. Contexto hardware do processo
   B. Contexto software do processo
   C. Tabela de interrupções
   D. A tabela de *signals* global do sistema.

B

b) [1,6v] Considere que um processo que executa o programa acima é colocado em execução e chega ao ciclo *while*. Enquanto está neste ciclo, o processo pode sair do estado "em execução" caso ocorram interrupções. Descreva duas dessas situações.

1) Situação 1:

Tipo de interrupção [**interrupção *hardware***][~~exceção~~][~~*trap*~~]

Causa da interrupção:
Utilizador premiu CTRL+C no teclado.
O que acontece ao processo quando retoma a execução?
Processo executa a função terminaComputacao.

2) Situação 2:

Tipo de interrupção [**interrupção *hardware***][~~exceção~~][~~*trap*~~]

Causa da interrupção:
Interrupção periódica do temporizador.
O que acontece ao processo quando retoma a execução?
Retoma a execução no ponto em que estava antes de ser interrompido.

Nota: outras respostas são possíveis e foram consideradas como corretas.

Grupo II [6 Val]

1. Considere uma arquitetura paginada de 32 bits e páginas de 4KBytes, com tabela de páginas de 1 nível. Num dado momento, o processo P1 está em execução. De seguida é apresentado um excerto da sua tabela de páginas.

Página	P	Prot	R	M	Base
0x00010	1	RW	0	1	0x08000
0x00011	1	RW	0	0	0x08001
0x00012	1	RW	1	0	0x0A004
0x00013	0	RW	0	0	0x0A004
0x00014	1	RW	1	0	0x0A005
0x00015	1	R	1	0	0x00010
0x00016	1	R	1	0	0x00F12
0x00017	1	R	1	0	0x00220

a) [1v] As páginas virtuais 0x00012 e 0x00013 têm a mesma base. Esta situação é correta? Se sim, apresente uma possível explicação. Se não, justifique.

Correto.
Possível explicação: a página virtual 0x00013 esteve presente em memória primária (no endereço real 0x0A004000) mas foi substituída em memória pela página 0x00012, daí terem a mesma base.

b) P1 fez alguns acessos a endereços virtuais na gama de páginas virtuais descrita pelo excerto acima. Abaixo descreve-se o que se passou em dois desses acessos. Para cada alínea, indique o endereço virtual que, ao ser acedido por P1, causou a situação descrita nessa alínea.

i) [1v] Acesso ao endereço físico 0x0A004088, completado sem originar nenhuma exceção.

0x00012008

ii) [1v] Ocorreu falta de página; núcleo transferiu página do disco para o endereço real 0x05555000; devolvida a execução a P1, ocorreu acesso ao endereço físico 0x05555222.

0x00013222

c) [1v] Suponha que o gestor de memória tem de escolher entre libertar o espaço em memória física ocupado pela página virtual 0x00010 ou pela página virtual 0x00011. Qual a melhor opção? Justifique.

Libertar o espaço ocupado pela página virtual 0x00011 é melhor. Como esta página não está modificada, não é preciso transferir o seu conteúdo para disco (ao contrário da página virtual 0x00010).

2. Considere o seguinte programa, que escreve o valor zero sobre algumas posições de um vetor de grande dimensão.

```
char *array = malloc(DIM);
int i;
for (i=0; i<DIM; i+=STEP) {
  array[i]=0;
}
```

a) [1v] Correndo o programa acima numa arquitetura com páginas de 1KB, monitorizou-se o desempenho do sistema quando STEP=2 e quando STEP=512. Uma das diferenças mais evidentes foi que a *TLB hit rate* mudou consideravelmente em cada experiência.
Com qual valor de STEP espera que a *TLB hit rate* seja superior? Justifique, idealmente apresentando uma estimativa da *TLB hit rate*.

O TLB *hit rate* será superior com STEP=2.
Com STEP =2, cada página virtual é acedida por 512 iterações consecutivas. Se o primeiro desses acessos implicar o carregamento da PTE respetiva a partir da tabela de páginas em memória, os restantes 511 acessos à mesma página já beneficiarão da presença da PTE em TLB. Ou seja, o TLB *hit rate* será de 511/512 (superior a 99%).
Com STEP=512, cada página é acedida 2 vezes consecutivas apenas, logo o TLB *hit rate* é reduzido para ½ (ou seja 50%).

b) [1v] Noutro conjunto de experiências, fixou-se o valor de STEP mas colocaram-se dois processos (em vez de apenas um) a executar o programa acima, competindo pelo mesmo CPU (*single core*). Espera que a *TLB hit rate* quando apenas corre um processo seja melhor, pior, ou idêntica a quando correm dois processos? Justifique.

A TLB *hit rate* piora quando há 2 processos.
Duas razões plausíveis (ambas foram consideradas como correctas):
- Quando P2 recebe CPU, a TLB é limpa, logo quando P1 recebe CPU de novo vai sofrer TLB *misses* adicionais.
- Quando P2 recebe CPU, os seus acessos vão inserir novas PTEs na TLB, substituindo PTEs de P1; logo, quando P1 recebe CPU de novo vai TLB *misses* adicionais.

Grupo III [6 Val]

Num sistema Linux existe um volume mantido com FAT-16 e outro volume com Ext3. A dimensão dos blocos em ambos os sistemas de ficheiros é de 1KByte. A dimensão de uma referência de bloco em ambos os sistemas é de 2 bytes.

1. [1,1v] No volume com Ext3, foi criado um novo ficheiro, cujo conteúdo inicial foi preenchido com 180KB de dados. No momento em que estas operações ocorreram, neste volume estavam livres os blocos 1200, 1201, 1202, e seguintes. Abaixo, preencha o conteúdo do vetor de índices de blocos do i-node deste novo ficheiro.

Entradas diretas												Ind. 1	Ind . 2	Ind . 3
1200	**1201**	**1202**	**1203**	**1204**	**1205**	**1206**	**1207**	**1208**	**1209**	**1210**	**1211**	**1212**		

2. [1,1v] Qual a dimensão máxima de um ficheiro em cada volume? Justifique apresentando os cálculos.

FAT-16: 2^{16} x 1KB

Ext3: (12 + 512 + 512^2 + 512^3) * 1KB

3. Ambos os volumes foram re-formatados para passarem a ter blocos de 2KBytes (ou seja, o dobro do tamanho). Este aumento do tamanho do bloco tem implicações:

   a. [0,5v] Na fragmentação interna?

      [**sim, aumenta**] [sim, reduz] [nenhuma implicação]

   b. [0,5v] No número de acessos à FAT/i-node Ext3 que têm de ser feitos para ler ou escrever num ficheiro grande?

      [sim, aumenta] [**sim, reduz**] [nenhuma implicação]

   c. [0,5v] Na dimensão da tabela de i-nodes do Ext3?

      [sim, aumenta] [sim, reduz] [**nenhuma implicação**]

   d. [0,5v] Na dimensão do mapa de blocos livres?

      [sim, aumenta] [**sim, reduz**] [nenhuma implicação]

4. As alíneas seguintes pretendem comparar diferentes opções de desenho relacionadas com sistemas de ficheiros. Para cada alínea, apresente uma **vantagem clara da primeira alternativa** citada nessa alínea.

    a. **[0,6v] Descritores de ficheiros mantidos em tabela de i-nodes** vs. **ficheiro totalmente descrito na entrada da diretoria em que existe**.

Diretoria mais compacta, logo a pesquisa de ficheiros nessa directoria fica mais eficiente.

ou

Suporte a *hard links.*

(Outras vantagens plausíveis foram consideradas correctas.)

    b. **[0,6v] Tabela de ficheiros abertos global** (além de tabelas de ficheiros abertos de cada processo) vs. **apenas tabelas de ficheiros abertos por processo** (sem tabela global).

Permite que 2 processos pai-filho partilhem o cursor de um ficheiro aberto pelo processo pai.

(Outras vantagens plausíveis foram consideradas correctas.)

    c. **[0,6v]** Sistema de ficheiros **em modo utilizador** vs. **em modo núcleo**?

Melhor robustez do sistema, pois eventuais erros no código do sistema de ficheiros passam a estar isolados no processo respetivo.

ou

Melhor portabilidade do sistema de ficheiros, pois pode ser instalado sem implicar alterações no núcleo do sistema operativo.

(Outras vantagens plausíveis foram consideradas correctas.)