Número: Nome:

LEIC/LETI – 2015/16 – 2º Exame de Sistemas Operativos

27 de Janeiro de 2016

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

Duração: 2h30

Grupo I [3 Val]

Considere o seguinte excerto de um programa. Assuma que a função f está implementada noutra parte do mesmo programa.

```
main ()
{
1    int pid;
2    int status;
3    int i = 0;

4    pid = fork ();

5    if (pid == 0) {
         i++;
6        f();
7        exit(0);
8    }
9    else if (pid > 0) {
10       sleep(1);
11       kill(pid, SIGKILL);
12       wait(&status);
13    } else {
14       /* ... */
     }
}
```

1. Este programa tem comportamentos diferentes consoante o tempo passado até a função f completar.
   a. [0,75 v] Explique quais são esses comportamentos e a condição que origina cada um.

   b. [0,5 v] Como pode o processo pai determinar qual o comportamento que ocorreu? Justifique.

2. [0,5 v] Em vez da chamada *sleep(1)*, a espera poderia ser implementada por um ciclo como:

```
while(k<NUM) k++;
```

Que diferenças encontra entre uma solução (sleep) e outra (ciclo while)?

3. [0,5 v] Quando a função *wait* retorna, que valor é observado pelo processo pai na variável *i*? Justifique.

4. [0,75 v] Para cada um dos seguintes elementos do contexto do processo pai, indique se é igual ou diferente no processo filho imediatamente após a criação deste.
   a. User identifier (UID)
   b. Group identifier (GID)
   c. Process identifier (PID)
   d. Parent process identifier (PPID)
   e. Conjunto de ficheiros abertos

5.

Grupo II [3,5 Val]

1. Considere o seguinte cenário que ocorre tipicamente em conferências com vários participantes aquando do "coffee break".
   - Existe uma mesa com um conjunto de taças e um jarro de café. Cada participante pega num taça e desloca-se para o jarro para a encher. O jarro fornece, para cada taça, uma dose fixa.
   - Em qualquer instante, apenas pode haver um participante a retirar uma taça da mesa, e apenas um participante a usar o jarro de café.
   - Existem empregados que repõem taças limpas na mesa e que repõem o café dentro do jarro.

   Considere o pseudo-código que se apresenta de seguida.
   Cada participante executa o código da função `beber_cafe()` e cada empregado executa as funções `repoe_tacas()` e `repoe_cafe()`.
   As funções `obtem_taca()`, `enche_taca()`, `poe_tacas_na_mesa()` e `enche_jarro()` têm a funcionalidade indicada pela sua designação, e não são bloqueantes.
   A parte superior da tabela abaixo apresenta as variáveis globais.

```
semaforo_t semTaca(NT);
semaforo_t semCafe(NC);
trinco _t trincoTaca(ABERTO);
trinco_t trinco_Cafe(ABERTO);
```

```
1.  beber_cafe() {
2.      esperar(semTaca);
3.      fechar(trincoTaca);
4.      obtem_taca();
5.      abrir(trincoTaca);
6.      esperar (semCafe);
7.      fechar(trincoCafe);
8.      enche_taca():
9.      abrir(trincoCafe);
10. }
```

```
11. repoe_tacas() {
12.     int i, T;
13.     fechar(trincoTaca);
14.     T=poe_tacas_na_mesa();
15.     abrir(trincoTaca);
16.     for (i=0;i<T;i++)
17.         assinalar(semTaca);
18. }
```

```
19. repoe_cafe() {
20.     int i, C;
21.     fechar(trincoCafe);
22.     C=enche_jarro();
23.     abrir(trincoCafe);
24.     for (i=0;i<C;i++)
25.         assinalar(semCafe);
26. }
```

   a. [0,5 v] Qual o significado do valor NT com que o semáforo semTaca é inicializado? Justifique.

b. [0,75 v] Qual a razão para, na função `repoe_tacas`, efectuar o ciclo invocando T vezes a chamada assinalar?

c. [0,75 v] Segundo esta implementação, há sempre espaço na mesa para mais taças?

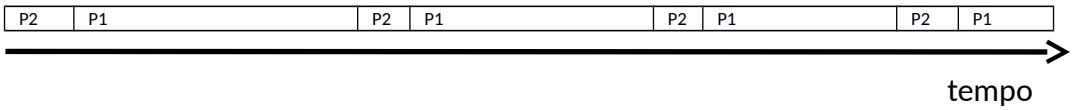
d. [0,75 v] Descreva uma situação real (com pessoas) que podia ocorrer caso as instruções das linhas 13 e 15 da funçao `repoe_taca`) não existissem.

e. [0,75 v] Diga se a situação seguinte pode ocorrer: *dois participantes pegam na mesma taça*. Se sim, explique porquê e apresente uma solução; senão, explique porquê indicando as linhas de código que o impedem.

Grupo III [3,5 Val]

1. [0,5 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.

2. 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:



   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. [0,5 v] Apresente possível uma razão para P1 ter maiores fatias de execução que P2 no exemplo acima.

b. [0,5 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.

c. Considere as comutações de processo, entre P1 e P2.
   a. [0,5 v] Dê 3 exemplos de elementos do contexto software que mudam quando se comuta de processo.

   b. [0,5 v] Dê 3 exemplos de elementos do contexto hardware que mudam quando se comuta de processo.

3. 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. [0,5 v] Indique, em pseudo-código, o que faz a linha 2.

b. [0,5 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 IV [3,75 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. [0,75v] 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. [0,75v] 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. 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. [0,75 v] 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) [0,75v] Indique quanta fragmentação interna (em bytes) existe nos blocos de dados deste ficheiro. Justifique.

ii) [0,75v] Indique que meta-dados associados a este ficheiro são mantidos em disco. Justifique.

Grupo V [2,75 Val]

1. Considere um sistema operativo do tipo Linux a correr sobre uma arquitetura paginada. Assuma que o computador em causa tem uma arquitectura paginada de memória virtual de 6 bits. Neste sistema, cada endereço virtual é composto em 2 bits (mais significativos) que indicam o n° de página e 4 bits (menos significativos) que indicam o deslocamento. Assuma que não existe TLB.

   a. [0,5 val] Qual a dimensão (em número de páginas) que deve ter a memória principal para que contenha todas as páginas virtuais do espaço de endereçamento de apenas um processo em qualquer instante? Justifique a sua resposta.

   b. [0,5 val] Qual a dimensão máxima do espaço de endereçamento de um qualquer processo (em bytes)? Justifique a sua resposta indicando o número máximo de páginas e a dimensão de cada página.

c. [0,5 val] Assuma agora, para esta questão apenas, que a dimensão física da memória principal tem uma dimensão superior ao valor máximo do espaço de endereçamento que indicou acima. Existe alguma vantagem nisso? Se sim, indique uma. Qualquer que seja a sua resposta, justifique.

d. [0,75 val] Tenha em conta noção de *working set* (espaço de trabalho) e que o computador em causa tem 2 processos (P1, P2) em execução. Assuma que o *working set* dos processos em causa é, respectivamente, 3 e 4 páginas. Diga se P1 se pode executar de forma eficiente. E o P2? Justifique a sua resposta ordenando os processos tendo em conta as *page-faults* que irão ocorrer e a noção de *working set*.

2. [0,5 val] Considere agora que o computador em causa (referido na questão anterior) tem uma TLB. Considere a seguinte frase e diga se concorda, ou não, justificando: "*quando ocorre um context-switch (i.e. a troca de um processo em execução por outro) o conteúdo da TLB é limpo*."

Grupo VI [3,5 Val]

1. Considere um sistema operativo do tipo Linux. Assuma que num processo P1, são redirecionadas as entradas/saídas da forma que se indica de seguida: i) *stdin* recebe dados de um pipe, ii) *stdout* envia dados para um socket.

   a. [0,5 val] Assuma que o pipe não tem nome, estando a extremidade de escrita num processo P2. Qual a relação entre P1 e P2? Justifique.

   b. [0,5 val] Assuma que o pipe tem nome. A extremidade de escrita pode encontrar-se num processo num computador remoto ? Justifique relacionando com o sistema de ficheiros.

   c. [0,5 val] Caso o *socket* suporte o modelo de ligação com diálogo (TCP/IP), qual a *system call* que o processo emissor deve invocar para fazer o pedido de estabelecimento de ligação?

d. [0,5 val] Tendo em conta a questão anterior, diga se o socket usado para fazer *accept* é o que será usado para receber dados? Justifique.

e. Considere um sistema operativo do tipo Linux e as respectivas caches do sistema de ficheiros, no qual foi criado um pipe com nome. Qual a cache mais relevante, do ponto de vista da minimização do tempo de execução, das operações seguintes quando usadas para criar e aceder ao pipe com nome:

f. [0,5 val] mkfifo.

g. [0,5 val] read.

h. [0,5 val] write.