

Sistemas Operativos, Exame 1

IST - LEIC-A/ LEIC-T/ LETI - 2017-2018
10 de Janeiro de 2017

- Todas as respostas devem ser dadas na folha de resposta.
 - Não pode sair da sala antes de passarem 60 minutos. Não é autorizada a utilização de telemóveis ou outros dispositivos electrónicos.
 - O exame tem a duração de 3 horas.
 - Em todas as respostas com código, pode omitir a verificação e tratamento de erros na chamada a funções.
-

1 Programação Com Tarefas Por Troca de Mensagens

Pergunta 1 (2 valores)

Considere um programa que pretende adivinhar uma palavra-passe da qual só tem a versão cifrada. Para isto é necessário cifrar várias palavras de um dicionário (de palavras conhecidas por serem comumente escolhidas por utentes pouco informados, como “qwerty” ou “benfica”) até encontrar uma palavra em claro (ou seja, não cifrada) cuja cifra seja igual à palavra-chave cifrada (daqui em diante designada por *palavra alvo*). Por simplicidade assuma que todas as palavras no dicionário, assim como a palavra alvo, têm no máximo 10 caracteres. As palavras do dicionário estão pré-armazenadas numa matriz em memória.

Pretende-se uma versão paralela deste programa, em que uma tarefa mestre cria N tarefas que vão testando X palavras em paralelo. O programa deve concretizar a seguinte lógica através da troca de mensagens:

1. A tarefa mestre cria N tarefa escravas, passando a palavra alvo como parâmetro na criação das mesmas.
2. Envia para cada escravo X palavras do dicionário (palavras diferentes para cada escravo).
3. Cada escravo testa as X palavras que recebeu. Se encontrar uma que, quando cifrada, corresponde à palavra alvo, envia-a ao mestre. Se não encontrar, envia uma string vazia ao mestre.
4. O mestre recebe as respostas dos N escravos.
5. Se um dos escravos encontrou a palavra-passe, a mestre imprime-a e retorna (da função *main*).
6. Se nenhum escravo encontrou a palavra-passe, a mestre volta ao ponto 2 do algoritmo e inicia uma nova iteração, enviando mais X palavras a cada escravo.

Complete o programa apresentado na folha de respostas, tendo em conta que:

- Deve recorrer a uma biblioteca de troca de mensagens com a seguinte interface:

```
int enviarMensagem(int tarefaOrig, int tarefaDest, void *msg, int tamanho);
int receberMensagem(int tarefaOrig, int tarefaDest, void *buffer, int tamanho);
```

que permite, respetivamente, enviar e receber uma mensagem da tarefa *tarefaOrig* para a tarefa *tarefaDest*, em que a tarefa mestre tem identificador 0 e as escravas são identificadas como 1, 2, 3, etc. Em caso de sucesso, ambas as funções retornam um inteiro que indica o número de bytes enviados/recebidos.

- O programa de cada escravo deve chamar a seguinte função auxiliar que, recebendo um vetor de x palavras e a palavra-chave alvo (cifrada), retorna um apontador para a palavra-chave em claro cujo valor cifrado corresponde à palavra alvo, ou NULL caso não encontre tal palavra-chave.

```
char *testa_x_palavras(int x, char *em_claro[], char *alvo);
```

- Deve omitir a verificação de erros na chamada às funções e assumir que o número de palavras no dicionário é múltiplo de $N \times X$.

2 Programação Com Tarefas Por Memória Partilhada

Considere uma biblioteca que permite a múltiplas tarefas partilharem um número natural (inteiro não negativo). O valor partilhado pode ser alterado pelas seguintes funções:

- `void incrementa()`, que incrementa 1 unidade ao valor partilhado.
- `void decrementa()`, que decrementa 1 unidade ao valor partilhado; caso o valor no início da chamada seja nulo, espera até que essa condição deixe de se verificar e finalmente decrementa a variável.

§

Considere a seguinte concretização das funções.

```
int valor = VALOR_INICIAL; //variável global com o valor partilhado

void decrementa() {
    while (valor == 0);
    valor --;
}

void incrementa() { valor ++; }
```

Usando esta concretização num programa com múltiplas tarefas concorrentes, registaram-se situações em que a variável partilhada chegou a valor negativo.

Pergunta 2 (1 valor) Descreva uma possível execução com 2 tarefas que leve a esta situação. No seu exemplo, assuma máquina *single-core* e que o valor inicial da variável partilhada é 1.

§

Considere agora seguinte concretização alternativa, que reorre a trincos lógicos e variáveis de condição.

```
int valor = VALOR_INICIAL; //variável global com o valor partilhado

mutex_t m;
condvar_t c;

void decrementa() {
    lock(m);
    if (valor < 1)
        wait(c, m);
    valor --;
    unlock(m);
}

void incrementa() {
    lock(m)
    valor ++;
    signal(c);
    unlock(m);
}
```

Mais uma vez, observaram-se situações em que a variável partilhada tomou valores negativos.

Pergunta 3 (1 valor) Apresente uma possível execução que ilustre o problema. De novo, assuma máquina *single-core*, 2 tarefas, valor inicial de 1.

§

Pergunta 4 (2 valores) Proponha uma solução alternativa que recorra a trincos lógicos e semáforos.

3 Programação com Processos

Os docentes de Sistemas Operativos pretendem desenvolver um programa que, entre um conjunto de projetos submetidos pelos alunos, verifica como estes terminam (que valor devolvem quando terminam com `exit`, ou se terminam abruptamente devido a um *signal*).

- Existem N projetos, armazenados no sistema de ficheiros local, já compilados. A localização do ficheiro executável de cada projeto é mantida no vetor `pathNames[]`. `pathNames[i]` contém o caminho de acesso do executável do projeto i .

- Para cada projeto, o programa de teste deve lançar um processo filho que executará esse projeto. Todos os projetos são executados sem quaisquer argumento de linha de comandos.
- Os projetos devem ser testados um por um. Ou seja, o processo do projeto $i + 1$ só deve ser lançado quando o processo do projeto i tiver terminado.
- Para cada projeto concluído, o programa de teste deve registrar (no vetor *result*) o estado de terminação devolvido pela função `wait` no argumento `status`.

Pergunta 5 (2 valores) Complete o programa na folha de respostas para concretizar os requisitos acima.

§

Os professores criaram uma conta de utilizador para testes (UID=`soteste`). Para testar um conjunto de projetos, o professor faz *log in* na máquina usando a conta `soteste`, abre uma consola de linha de comandos e nessa consola lança o programa de teste. Em dado momento, o programa de teste lança um novo processo que executa o seguinte ficheiro executável de um projeto:

```
-rwxr-xr-x 1 rafael alunos 1436 Oct 11 22:47 projeto12
```

Pergunta 6 (0,5 valores) O processo tem permissões para executar este ficheiro? Se sim, qual o EUID (UID efetivo) do processo filho ao executar o ficheiro? Se não, justifique.

Noutro momento, o programa de teste lança um novo processo que executa o seguinte ficheiro executável de um projeto:

```
-rwx----- 1 manuel alunos 1446 Oct 11 23:59 projeto12
```

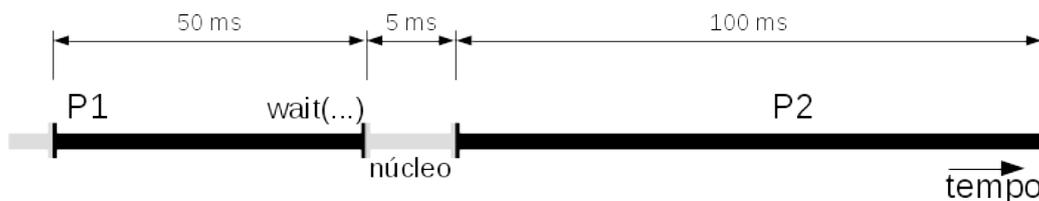
Pergunta 7 (0,5 valores) O processo tem permissões para executar este ficheiro? Se sim, qual o EUID (UID efetivo) do processo filho ao executar o ficheiro? Se não, justifique.

§

Pergunta 8 (1 valor) Um dos professores também tem conta *super user* (*root*) nesta máquina e por vezes corre o programa de testes usando essa conta. Concorda com esta prática? Justifique apresentando uma vantagem ou desvantagem.

4 Gestor de Processos

Numa máquina Unix com 1 CPU, observou-se a seguinte utilização do CPU. P1 e P2 são processos, sendo que P1 se bloqueou ao chamar a função `wait`.



Pergunta 9 (1 valor) Quais as últimas instruções assembly executadas antes da transição para a rotina do núcleo e antes da transição da rotina do núcleo para a execução de P2?

Pergunta 10 (1 valor) No início da execução ilustrada acima, ambos os processos tinham prioridades iguais. Assuma também que ambos os processos não voltaram a executar-se antes do escalonador do Unix recalculas as prioridades. Após o recálculo, qual será mais prioritário, P1 ou P2? Justifique recorrendo à fórmula de recálculo de prioridades usada pelo escalonador do Unix.

5 Gestão de Memória

Considere um sistema com paginação, em que as páginas possuem 4Kbytes. Considere a seguinte tabela de páginas de um processo:

Página	Bit Presença	Protecção	Acedido (R)	Dirty (M)	Base
0	0	-	0	0	0x000
1	1	RW	1	1	0x0AB
2	1	R	1	0	0x0FA
3	1	RW	0	0	0x053
4	0	RW	0	1	0x031
5	0	RW	0	0	0x032
6	0	W	1	1	0x033

Considere também que, caso seja necessário carregar uma página deste processo para memória, esta será colocada na trama (página física) 0x045.

Pergunta 11 (1 valor) Preencha a tabela com a tradução entre endereços reais virtuais e endereços reais que se encontra na folha de respostas.

Para cada acesso indique:

- Se ocorreu uma falta de página (coloque um “S”(im) ou um “N”(ão) na coluna FP).
- Qual o endereço físico gerado.
- Nos casos em que um endereço físico não é gerado, a causa para o erro.

§

Considere exactamente a mesma tabela de páginas, no estado representado anteriormente.

Pergunta 12 (0,5 valores) Se necessitar de retirar uma página a este processo, qual é a melhor candidata para sair de memória? Justifique.

Pergunta 13 (0,5 valores) Se necessitar de retirar uma segunda página a este processo que página retiraria (adicionalmente à que indicou na questão anterior)? Justifique.

§

Pergunta 14 (1 valor) Considere que o processo acima chamou a função fork no momento em que tinha a tabela de páginas tal como se encontra acima no enunciado. Preencha o estado da tabela de páginas do processo filho imediatamente após este ser criado.

§

Pergunta 15 (1 valor) Indique o que entende por espaço de trabalho (*working set*) de um processo.

6 Sistemas de Ficheiros

Considere que a diretoria raiz de um dado sistema de ficheiros Unix possui o seguinte conteúdo.

Inode	Tamanho Entrada	Tamanho do Nome	Tipo	Nome
2	12	1	2	.\0\0\0
2	12	2	2	..\0\0
11234	12	3	2	tmp\0
11111	12	5	1	f.txt\0\0\0

Considere que existe uma outra diretoria com o seguinte conteúdo:

Inode	Tamanho Entrada	Tamanho do Nome	Tipo	Nome
11234	12	1	2	.\0\0\0
2	12	2	2	..\0\0
22222	12	5	1	f.txt\0\0\0

Pergunta 16 (0,5 valores) Qual é o inode do ficheiro “/f.txt”? Justifique

Pergunta 17 (0,5 valores) Qual é o inode do ficheiro “/tmp/f.txt”? Justifique

§

Considere que não existem nem nenhuns inodes em memória nem nenhum bloco na cache de blocos. Desta forma será necessário trazer inodes para memória e blocos para memória sempre que for necessário aceder a alguma informação no disco.

Pergunta 18 (1 valor) Diga quantos inodes e quantos blocos é necessário trazer para memória para ler o primeiro byte do ficheiro “/tmp/f.txt”. Justifique.

§

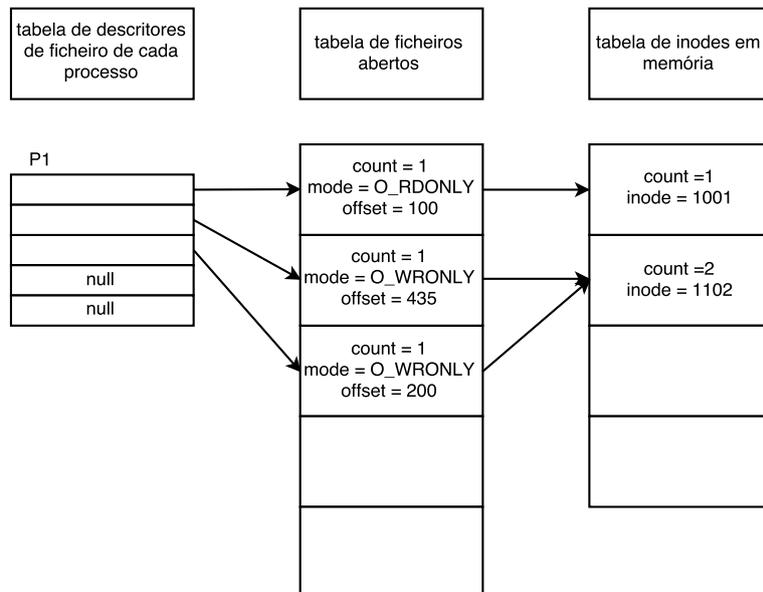
Considere um sistema ficheiros do Unix do tipo ext3 em que os inodes possuem uma tabela com 15 apontadores e os blocos 4Kbytes. Considere um ficheiro acabado de criar e no qual ainda não se escreveu nada. Considere que os 5 primeiros blocos livres em disco são os seguintes: 1000, 2000, 2500, 1200, 2222. Considere que nesse ficheiro, ao qual pode aceder usando o descritor “fd”, se executa a seguinte sequência de comandos:

```
lseek(fd, 10000, SEEK_SET);  
write (fd, buff, 10);  
close (fd);
```

Pergunta 19 (1 valor) Indique qual a conteúdo da tabela de índices do inode correspondente a esse ficheiro após a execução de sequência acima.

§

Considere o seguinte estado das estruturas em memória mantidas pelo sistema de ficheiros.



Pergunta 20 (1 valor) Considere que o processo P1 executa com sucesso o seguinte excerto:

```
int pipe_fd [2];  
  
pipe(pipe_fd);  
pid = fork();  
if (pid > 0) open("/tmp/f.txt", O_RDONLY);
```

Desenhe **no espaço reservado na folha de respostas** as alterações nas estruturas do sistema de ficheiros mantidas em memória.

Sistemas Operativos, Exame 1, 10 de Janeiro de 2018

IST - LEIC-A/ LEIC-T/ LETI - 2017-2018

Folha de Respostas (1/4)

Número:

Nome:

§

Programação com tarefas por troca de mensagens

Pergunta 1

```
#define X 10
#define N 10
#define CHANNELSZ 20
#define NPALAVRAS 5000
#define PASS.SIZE 10

//Pré-preenchido com palavras comuns
char dicionario[NPALAVRAS][PASS.SIZE];

typedef struct {
    int id;
    char alvo[PASS.SIZE];
} argsEscrava_t;

/*-----
| Function: fn_escrava
-----*/
void *fn_escrava(void *a) {
    argsEscrava_t *arg = (argsEscrava_t *) a;
    int myid = arg->id;
    char *alvo;

    char recebidas[X][PASS.SIZE];
    char *s;

    while (1) {

        /* Recebe X palavras e coloca-as em 'recebidas' */

        s = testa_x_palavras(X, recebidas, alvo);

        /* Responde à mestre */

    }

    return 0;
}
```

```

/*-----
| Function: main
/*-----*/
int main (int argc, char** argv) {
    argsEscrava_t escrava_args[N];
    pthread_t escravas[N];

    int t, i;

    char *alvo = argv[1]; //testes aos argumentos omitidos

    /* Inicializa biblioteca de troca de mensagens
    (capacidade do canal, numero de tarefas comunicantes) */
    inicializarMPLib(CHANNELSZ, N+1);

    for (t=1; t <= N; t++) {

        pthread_create(&escravas[t-1], NULL,
                      ,
                      );

    }

    //i indica o índice da próxima palavra do dicionário a enviar
    i = 0;

    while (i < NPALAVRAS) {

        /* Envia as próximas X palavras em mensagem */
        enviarMensagem (
            ,
            &dicionario[i], X*PASS_SIZE);
        i = i + X;

    }

    printf ("Palavra-chave_ nao_encontrada\n");
    return 1;
}

```

Folha de Respostas (2/4)

Número:

Nome:

§

Programação com tarefas por memória partilhada

Pergunta 2

Pergunta 3

Pergunta 4

Pergunta 5	
------------	--

```
/* pré-preenchida com as localizações de cada executável */
char *pathNames[N];
```

```
int results[N];
```

```
int main(..) {
```

```
    int i;
    int status;
```

```
    for (i=0; i<N; i++) {
```

```
        results[i] = status;
```

```
    }
```

```
    imprimeResultados(results);
    return 0;
```

```
}
```

Folha de Respostas (3/4)

Número:

Nome:

§

Pergunta 6	
------------	--

Pergunta 7	
------------	--

Pergunta 8	
------------	--

§

Gestor de Processos

Pergunta 9	
------------	--

Pergunta 10	
-------------	--

Pergunta 11				
Acesso	Endereço Virtual	FP	Endereço Real	Excepção lançada
Leitura	0x001345			
Leitura	0x002AFC			
Leitura	0x001ABD			
Escrita	0x003A4F			
Leitura	0x004125			
Escrita	0x002000			

Pergunta 12	
-------------	--

Pergunta 13	
-------------	--

Pergunta 14					
Página	Bit Presença	Protecção	Acedido (R)	Dirty (M)	Base
0					
1					
2					
3					
4					
5					
6					

Pergunta 15	
-------------	--

Folha de Respostas (4/4)

Número:

Nome:

§

Sistema de Ficheiros

Pergunta 16	
Pergunta 17	
Pergunta 18	
Pergunta 19	

