



90 minutos

Nome _____

nº _____

Regras

Qualquer tentativa de fraude conduzirá à anulação da prova para todos os intervenientes.

Consulta: Apenas em papel. Não são permitidos meios electrónicos (calculadora, telemóvel, ...).

As respostas devem ser obrigatoriamente efectuadas na folha fornecida.

1 Questão (85%)

Pretende-se criar um conjunto de processos que contem a ocorrência de diferentes números (de 0 a 9) numa matriz. A matriz a considerar encontra-se em memória partilhada e é inicializada por um outro processo (processo gerador). Cada processo conta um número específico que lhe é atribuído na altura da sua criação. Depois de todas as contagens serem efectuadas, o processo gerador volta a inicializar a matriz com novos números de 0 a 9. Este processo é repetido 3 vezes, após as quais todos os processos são terminados e os recursos usados eliminados.

- Mapeie a memória partilhada, onde a matriz e as variáveis auxiliares estão guardadas.
- Crie todos os processos necessários (são no total 11 - um por número de 0 a 9 a procurar e um adicional para gerar os números da matriz).
- O processo gerador inicializa a matriz com números de 0 a 9 e dá a cada matriz um número sequencial diferente.
- Cada processo apenas percorre cada matriz uma vez. Para isso, tem de verificar previamente o número sequencial da matriz existente em memória partilhada para não repetir contagens da mesma matriz. Se já tiver contado a matriz existente e o número de matrizes ainda não tiver chegado a 3, dorme durante 1s.
- Os processos, após cada contagem, escrevem no ecrã o número de ocorrências encontradas.
- O processo gerador está bloqueado até que todos os outros processos terminem as contagens.
- O processo principal espera que todos os processos filho terminem, limpa todos os recursos usados e sai.
- Declare todas as variáveis que precisa no início e use todos os mecanismos de sincronização necessários a um bom funcionamento do programa.

Complete o código que se segue, escrevendo nos locais a cinzento.

(...)

```
#define TOTAL_MATRICES 3 // número de matrizes a calcular
```

```
// declaração de variáveis globais
```

```
typedef struct mem_shared {  
    int matrix[10][10];  
    int n_processed;      // n.o de algarismos já processados  
    int matrix_number;    // n.o identificador da matriz gerada – cada matriz é gerada  
                        // com um n.o sequencial diferente  
} mems;
```

```
mems* mems_ptr;
```

```
// Função a ser executada pelo processo gerador
```

```
}
```

```
// Função a ser executada pelos processos que contam as ocorrências dos algarismos
```

```
}
```

Nome:_____

Número:_____

```
int main(void) {
    // inicializações

    // Criação e mapeamento da memória partilhada

    // Inicializar variáveis em memória partilhada
    mems_ptr->n_processed=0;
    mems_ptr->matrix_number=0;

    // Criar o processo gerador e os processos que vão efectuar a contagem
    //(cada processo vai contar um número de 0-9)

    // esperar pelo fim dos processos

    // remover recursos

}
```

2 Questão (15%)

Imagine dois processos que comunicam entre si através de um pipe e implementam uma arquitetura Produtor/Consumidor. De forma trocarem apenas a informação necessária entre eles, foi definida a estrutura mensagem.

```
struct mensagem {  
    int      id_produto;      /* Identificação do produtor      */  
    int      size;           /* Tamanho do vector de dados     */  
    double*  dados;          /* Tabela dinamicamente reservada com os dados */  
};
```

O processo Produtor gera os valores e reserva o espaço de memória para o vector dados da estrutura mensagem através de um malloc. Depois escreve os dados gerados nesse vector e escreve a estrutura mensagem no pipe. No entanto, quando o processo Consumidor tenta obter os valores do vector dados ocorre um Segmentation Fault. Porquê?