

- Não é permitida a consulta de livros ou de apontamentos.
- Não se esclarecem dúvidas durante a prova. Se tiver dúvidas, indique na folha de teste a sua interpretação.
- Utilize uma caligrafia legível.
- Consultar a folha em anexo sobre sintaxes e semântica que deve utilizar para responder às alinhas.

### Grupo B Funcionamento dum Sistema Operativo (8 Valores)

1. O seguinte programa de Assembler devia escrever o string de mensagem (msg) para o canal de output padronizado de erro (Nota que STDERR\_FILENO é o valor 2), por defeito ligado ao ecrã e depois devolver o número de bytes do string para o processo que o invoca. No entanto está errado ! Deve corrigir o programa reescrevendo as linhas, movq, depois do \_start, cada uma tem um erro. Deve consultar a folha em anexo com a tabela das Chamadas ao Sistema relevantes **(2)**

```
.data      #initialized data section
msg:
    .ascii "SisOps\n"
    len = . - msg      #number of bytes do string
.text
    .global _start
_start:
    movq $11, %rax
    movq $1, %rdi
    movq $msg, %rdx
    movq $len, %rsi
    syscall
    movq $06, %rax
    movq $0, %rdi
    syscall
```

2. Considerando que agora o programa está bem escrito explique qual o output em baixo e explique em pormenor o que faz cada um dos comandos (Bash Shell) seguintes **(2)**

```
> cc -c test.S
> ld -o test test.o
> ./test
> echo $?
```

3. O que acontece ao programa na linha "syscall" ? - quer dizer como é que um sistema operativo moderno (por exemplo Linux) processa e executa a chamada ao sistema e em que "modo" de execução as instruções do processo vão ser executados e se existem interrupções e de que tipo? Deve explicar em pormenor **(4)**

### Grupo B Execução de Processos e Programas (4 Valores)

4. O programa seguinte,

```
#include<stdio.h>
int main(){
    fputc('o', stdout);fputc('l', stdout);fputc('a', stdout);fputc('\n', stdout);
}
```

quando executado em conjunto com **strace** tem o output seguinte em baixo. Explique este output e o que está a acontecer em cada linha **(2)**

```
execve("./a.out", ["/a.out"], ... ) = 0
openat( ... , "/lib/x86_64-linux-gnu/libc.so.6", ... ) = 3
close(3)
write(1, "ola\n", 4 )
```

5. Qual é o output do seguinte programa? O output é determinístico independentemente do algoritmo de escalonamento? Justifique! (Deverá mostrar o *trace* do funcionamento do programa). **(2)**

```
char * args[2]={ "date",NULL};
int  status, pid, x = 1;
pid = fork();
if (0==pid ) {
    x++;
    pid = fork();
    if (0==pid) execvp( *args, args );
    else {
        x++;
        wait(&status);
    }
}
else {
    wait(&status);
    x--;
}
printf("x=%d\n",x);
```

### Grupo C Escalonamento (4 valores)

6. Faça o diagrama temporal da execução dos processos indicados na tabela abaixo seguindo o algoritmo de escalonamento por prioridades com preempção. Calcule o tempo médio de circulação (turnaround). **(2)**

processo	tempo de chegada	prioridade	duração
P1	1	5	2
P2	1	3	4
P3	2	1	2
P4	6	2	2

7. Considere um sistema de tempo real com 3 processos periódicos (A, B e C) com períodos de A (4) B (6) e C (8) unidades do tempo e tempos de execução/CPU de A (1), B(2) e C(3) unidades respetivamente. **(2)**
- (i) Faça o diagrama de Gantt até 12 Unidades de tempo usando o algoritmo de Escalonamento "Rate Monotonic". Existe algum "deadline miss" (falha do prazo temporal) ? Explique a sua resposta.
- (ii) Para este sistema poderá existir um escalonamento competente e viável ou não? Explique.

### Grupo D Exercício Prático (4 valores)

8. Escreva a função miniFile que remove todo o espaço em branco dum ficheiro, nomeadamente os caracteres de tab, espaço e novalinha, escrevendo o resultado para um novo ficheiro

*Protótipo:* int miniFile ( char \*entrada, char \*saida, int blockSize);

A função deverá abrir o ficheiro (parâmetro entrada) em modo de apenas leitura e ler o ficheiro para memória em blocos máximo dum tamanho especificado (parâmetro blockSize). Os caracteres validos devem ser escritos num ficheiro (saída) com permissões de leitura e escrita para o próprio e também em blocos de tamanho máximo de "blockSize".

Em caso do erro a função deverá devolver -1 e em caso de sucesso zero. Não se esqueça de alocar e libertar memória adequadamente.

*Exemplo Utilização:* int status = miniFile ( "webfile.js", "webfile.minijs" , 4096 );

Não é necessário especificar as bibliotecas padrão (#include <fcntl.h> etc.) no seu programa.

Escreva APENAS o código da função.

A função pode ser escrita em C ou em Ocaml.

Dica : Considera a utilização de dois Arrays – um para a leitura que contém os dados originais e um outro para onde estão copiados apenas os caracteres válidos para serem escritos no ficheiro de saída.