

Pergunta 1

Não respondida

Nota: 1,00

1. (13%) Numa directoria existem os seguintes ficheiros, tal como mostrado com `ls -l` (atenção aos detalhes)

```
-rw----- 1 sofia sofia    13 Feb 24 15:38 dados.txt  
-rwsr-xr-x 1 toze  toze  16704 Feb 24 15:39 faz
```

O conteúdo do ficheiro `fich.txt` é apenas a seguinte linha
`texto oculto`

O código do programa `faz` é o que está à direita
(excluindo `#includes`)

Indique e explique o que aparece e porquê quando o
utilizador `sofia` executa nessa directoria

```
$ ./faz dados.txt
```

```
int main(int argc, char * argv[]) {  
    char texto[1000] = { 0 };  
    strcpy(texto, "Principio");  
    int fd = open(argv[1], O_RDONLY);  
    read(fd, texto, 999);  
    printf("%s\n", texto);  
    return 0;  
}
```

A resposta é dada sob a forma de fotografia, submetida num link para o efeito (não é no teste). Só pode fotografar a resposta quando lhe derem ordem.




2. A função `fork()` é uma função sistema Unix. Considere que é invocada no contexto do processo utilizador PID 321.

a) (8%) Explique tudo o que está envolvido desde que a função é invocada até ser atingida a primeira instrução dessa função. Não está em questão o que é que a função faz, mas sim os mecanismos envolvidos na sua invocação.

b) (5%) indicação quais os problemas envolvidos nas chamadas às funções sistema e como são resolvidos.

A resposta é dada sob a forma de fotografia, submetida num link para o efeito (não é no teste). Só pode fotografar a resposta quando lhe derem ordem.



3. (14%) A aplicação do princípio de localidade de referência ajuda ou prejudica a gestão de memória? Em que medida? Explique e indique, justificando, o tipo de gestão de memória mais afectado.

A resposta é dada sob a forma de fotografia, submetida num link para o efeito (não é no teste). Só pode fotografar a resposta quando lhe derem ordem.

A large empty rectangular box with a thin black border, intended for a user to take a photograph of their answer. In the bottom right corner of the box, there is a small, faint icon of a camera or a similar symbol.

Pergunta 1

Não respondida

Nota: 1,00

(5%) Utilizando apenas uma linha de comandos, coloque no ficheiro "nomes.txt" o *login (username)* dos utilizadores do sistema que utilizam a *shell bash* e que têm o nome próprio começado por uma letra minúscula. Esta informação deve estar ordenada por ordem crescente do grupo primário do utilizador.

A resposta é dada sob a forma de fotografia, submetida num link para o efeito (não é no teste). Só pode fotografar a resposta quando lhe derem ordem.



(25%) Considere vários programas existentes que têm o seguinte comportamento: quando recebem o sinal SIGINT respondem a quem lhe enviou o sinal com o sinal SIGUSR1 no espaço de 3 segundos. Se não efetuarem essa resposta, é porque deixaram de funcionar.

Desenvolva a aplicação `mymonitor` que lance programas como os referidos acima e monitorize o seu funcionamento enviando-lhes o sinal SIGINT a cada NTEMP segundos (NTEMP é um `#define`). Os processos que não respondem no tempo previsto devem ser terminados, seguidos da escrita na consola de “Matei o programa X”, onde X é substituído pelo nome do programa que foi terminado. A aplicação termina após N ciclos de monitorização (sendo N = argumento `op1`), terminando também, de forma abrupta, os programas que ainda estiverem a funcionar nessa altura. A aplicação é lançada como se mostra no exemplo abaixo. Assuma que o número de programas nunca é superior a 10.

```
#define NTEMP=9
```

```
$ ./mymonitor op1 prog1 prog2 prog3 prog4 prog5
```

A resposta é dada sob a forma de fotografia, submetida num link para o efeito ([não é no teste](#)). Só pode fotografar a resposta quando lhe derem ordem.



Pergunta 1

Não respondida

Nota: 1,00

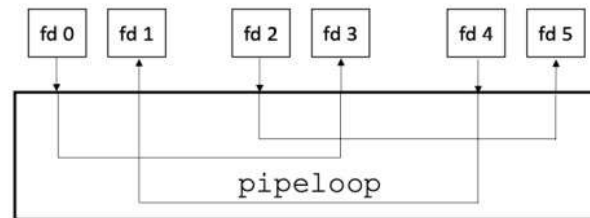
(5%) Coloque no ficheiro fic.txt na sua *homedir* o nome do segundo menor ficheiro que lhe pertence de entre os que estão na directoria */opt*. A sua solução deve funcionar a partir de qualquer ponto do sistema de ficheiros e não pode assumir *usernames* específicos.

A resposta é dada sob a forma de fotografia, submetida num link para o efeito (não é no teste). Só pode fotografar a resposta quando lhe derem ordem.



Escreva um programa `pipeloop` que recebe como argumento da sua função `main()` o número de pares (entre 3 e 12) de descritores abertos, relativos a pipes bloqueantes, herdados do processo pai (portanto tenha em atenção que para este programa os três primeiros descritores não têm a conotação usual de `0-stdin/1-stdout/2-stderr`). De cada par de descritores abertos o primeiro corresponde a uma extremidade de leitura e o segundo a uma extremidade de escrita. O trabalho do programa `pipeloop` resume-se a enviar tudo o que receber no extremo de leitura de um par para o extremo de escrita do par seguinte. A informação lida do último par é escrita no primeiro par.

Por exemplo, com a linha de comando `pipeloop 3` o programa assume que tem disponíveis os pares de descritores (0, 1), (2, 3) e (4, 5), sendo de leitura os descritores 0, 2 e 4 e de escrita 1, 3 e 5. O programa deve processar a informação assim que disponível.



Escreva o programa completo. Não pode recorrer a *threads*. Não especifique `#include` na sua solução.

(25%) A resposta é dada sob a forma de fotografia, submetida num link para o efeito (não é no teste). Só pode fotografar a resposta quando lhe derem ordem.

