

Sem consulta
Duração 45 min + 5 tol.

1. Explique por que razão/razões as directorias /tmp e /home são normalmente mapeadas em dispositivos diferentes do resto do sistema de ficheiros.

2. Explique i) o que é e ii) para que serve o bit setuid e iii) dê um exemplo habitual de uso deste bit.

3. Observe o código do programa “fazisto” à direita. Assumindo que não há erros nas chamadas às funções sistema, explique i) o que faz o programa e ii) o que aparece no ecrã.

```
int main() {
    int i=0;
    while (i < 3) {
        fork();
        i++;
        execl("fazisto", "fazisto", NULL);
    }
    printf("valor %d", i);
    return 0;
}
```

4. Considere que num determinado sistema são lançados os processos A,B,C,D nos instantes 0,1,3,4 (em segundos), respectivamente. Estes processos têm a seguinte duração em segundos, respectivamente: 5,1,4,2. Os processos são do tipo *CPU bound*.

- Considere que lhe dão a hipótese de gerir o escalonamento do sistema pelo algoritmo *Round Robin* ou pelo *Shortest Remaining Time* (ou um ou outro). Qual deles é o melhor para a situação indicada considerando como factor de decisão o tempo médio de espera?
- Explique quais as vantagens e desvantagens do algoritmo *Round Robin* face ao *Shortest Remaining Time* (genericamente: não tem a ver com a alínea anterior).

5. Normalmente as funções sistema têm capacidade de efectuar operações vedadas às funções biblioteca. Identifique e explique os mecanismos (hardware/software) envolvidos nesta diferença de capacidade (por outras palavras: “como funciona isso?”)

6. Num sistema com memória paginada encontra-se a correr o processo A cuja tabela de páginas é a que se encontra à direita. O sistema é de 32 bits, 12 dos quais usados para o deslocamento.

Diga

- Qual o tamanho da página neste sistema
- Qual o endereço real manipulado quando é executada uma instrução que pretende guardar o valor 123 no endereço virtual 4098?

Índice	P	Prot	Base	R	M
0	1	RWX	32768	0	0
1	1	RWX	8192	0	1
2	1	RWX	16384	1	0
3	0	RWX	0	1	1

1. [12,5%] Linha de comandos / Bash

Utilizando apenas uma linha de comandos de Unix e sem recorrer a ficheiros temporários (exceto se explicitamente indicado):

- a) (5%) Sem utilizar o mecanismo *sudo*, guarde no ficheiro **listaDir.txt** a lista de todas as diretorias do sistema de ficheiros. Os eventuais erros de acesso deverão ser guardados num ficheiro de nome **logErr.txt**.
- b) (7,5%) Apresente no ecrã **o número** de utilizadores registados no sistema que possuam a sua diretoria pessoal em **/home** e em simultâneo tenham a *Bash Shell* como interpretador de comandos.

2. [87,5%] Desenvolvimento em C para Unix

Pretende-se um sistema para gerir as descolagens de aviões num aeroporto. Cada avião é identificado por um conjunto de 6 caracteres e existem 5 controladores aéreos para estes aviões. O aeroporto tem 10 pistas, identificadas por um carácter de A a J, que podem estar abertas ou encerradas, e cada pista tem uma fila de espera com capacidade para 6 aviões. A cada avião corresponderá um controlador que irá gerir a sua descolagem. A atribuição de um avião a um controlador é feita por um outro sistema, devendo por isso assumir que cada controlador sabe os aviões que lhe estão atribuídos e que só exerce ações sobre estes.

Cada controlador utiliza uma aplicação cliente no seu terminal Unix para interagir com a central (esta central não é mais do que a aplicação servidor) para saber a situação atual no aeroporto e o que deve fazer. Existe um funcionário coordenador no aeroporto que interage diretamente com a central (“diretamente” = com a consola do servidor) segundo a lógica de comandos escritos. Os clientes dos controladores interagem com a central essencialmente por *named pipes* e a interação entre um controlador e o seu cliente segue também a lógica de comandos escritos. A comunicação entre o avião e o seu controlador não entra neste enunciado.

Um controlador pode executar as seguintes ações na sua aplicação cliente:

- Solicitar à central do aeroporto uma pista para um determinado avião através do comando **pista <avião>**. A central atribui ao pedido a 1ª pista que encontrar que não esteja encerrada e que tenha espaço na sua fila de espera. Após o pedido ser processado, é enviada uma mensagem para todos os controladores a indicar o estado atual das 10 pistas. A central nunca responde a um único cliente; em vez disso difunde a todos a situação actual resultante do processamento de um pedido (pistas abertas ou encerradas, ocupação das respetivas filas de espera). O controlador deverá analisar as mensagens e extrair a informação relevante para ele: perceber se um seu pedido anterior foi concretizado verificando se o avião em causa foi atribuído à fila de espera de uma das pistas. Se o avião não se encontrar em nenhuma fila o controlador deve assumir que o pedido não foi concretizado e tomar a decisão que entender (o sistema não se preocupa com este cenário);
- Se um controlador verificar que um dos seus aviões se encontra no topo de uma fila de espera pode então avisar o piloto que este deve descolar e, quando a descolagem terminar, regista esta ação na central através do comando **descolou <avião>**. A central ao receber este comando simplesmente retira o avião da fila de espera (passando o próximo avião ao topo da fila) e envia uma mensagem para todos os controladores a indicar o novo estado das 10 pistas. Se o avião enviado no pedido não estiver no topo da fila de espera de nenhuma pista o comando é simplesmente ignorado.

Quando um avião atinge o topo da fila de espera da sua pista este tem 10 minutos para descolar. Se ao fim destes 10 minutos a central ainda não tiver recebido o comando de descolagem para este avião a pista é automaticamente encerrada, recebendo todos os controladores essa informação. Quando encerrada uma pista não pode receber novos aviões na sua fila de espera nem novas descolagens. A pista só pode ser aberta novamente pelo coordenador da central através do comando **abrir <pista>**, voltando esta ao seu funcionamento normal.

Identifique sempre a alínea e subalínea da sua resposta. Justifique devidamente todas as respostas. Seja objetivo e detalhado.

- a) (20%) Quanto à arquitetura da solução para o sistema (central e clientes) indique, complementando a resposta textual sempre que possível com diagramas:
 - i) [5%] Estruturas de dados necessárias, incluindo as que vão circular nos *named pipes* (descrição + linguagem C).
 - ii) [5%] Quais os *named pipes* necessários, identificando a abordagem para os seus nomes, quem os cria, lê e escreve.
 - iii) [5%] Mecanismos de comunicação concorrente e sincronização a usar no servidor (*select()*, *threads*, *mutexes*, sinais, etc.).
 - iv) [5%] Qual a abordagem que propõe para que a central encerre as pistas quando os aviões não descolam em 10 minutos.

b) (67,5%) Implemente as seguintes componentes do servidor:

i) [17,5%] Código responsável pelo atendimento simultâneo dos controladores e do coordenador, incluindo a lógica dos *named pipes*, o mecanismo de comunicação concorrente e a leitura dos comandos (não deve implementar nenhum comando, mas deve indicar através de comentários onde cada um deles seria implementado).

ii) [17,5%] Código responsável pela lógica do comando **pista <avião>**.

iii) [17,5%] Código responsável pela lógica do comando **descolagem <avião>**, incluindo a lógica necessária para o encerramento automático da pista caso o novo avião que passa para o topo da fila de espera não descole em 10 minutos.

iv) [15%] Código responsável pela lógica do comando **abrir <pista>**.

Nota 1: Para situações não contempladas no enunciado deve descrever e realizar soluções coerentes e sensatas que não aumentem a complexidade da solução final mas também não removam partes da questão. Devem ser visíveis todos os aspectos relacionados com a alínea a): operações de comunicação concorrente, criação de *threads*, envio e tratamento de sinais, gestão de *mutexes*, etc..

Nota 2: Para simplificar a sua resolução deve evitar desenvolver lógica de manipulação de estruturas de dados, interpretação de *strings*, etc. Em seu lugar deve considerar a chamada a funções que realizem essas tarefas e descrever textual e sumariamente a sua especificação (o que recebem, retornam e processam), assumindo que se encontram codificadas algures.

ATENÇÃO: Não se esqueça de responder à pergunta 1 (12,5%).