

1. Explique o que entende por Processos em Sistemas Operativos.

Um processo é uma instância de um programa que está a ser executado

2. Considere as duas implementações de mecanismos de comunicação entre processos: memória partilhada, objeto de comunicação do sistema.
 - a) Compare-as tendo em conta a complexidade da programação da sincronização. Justifique sucintamente.

A utilização de memória partilhada exige uma maior complexidade na programação da sincronização, uma vez que os processos acedem simultaneamente à mesma área de memória. Por outro lado, os objetos de comunicação do sistema, como pipes, filas de mensagens ou sockets, já incluem mecanismos de sincronização integrados, simplificando o desenvolvimento e reduzindo a probabilidade de erros relacionados com concorrência.

- b) Compare-as em termos de eficiência. Justifique sucintamente

Em termos de eficiência, a memória partilhada é geralmente superior, pois permite que os processos comuniquem diretamente através de uma zona comum da memória, sem necessidade de cópia de dados ou envolvimento contínuo do sistema operativo. Isto reduz significativamente a latência e o overhead, especialmente quando se trata de grandes volumes de dados. Em contraste, os objetos de comunicação do sistema requerem múltiplas cópias de dados entre buffers e fazem uso intensivo de chamadas ao sistema operativo.

3. O que entende por Escalonamento?

Escalonamento é o processo de decidir qual dos processos deve utilizar o processador num dado momento. Como normalmente existem mais processos do que CPUs disponíveis, o escalonador tem de gerir o tempo de CPU de forma eficiente, garantindo que todos os processos têm uma oportunidade de execução, deve respeitar prioridades e maximizar o desempenho global do sistema.

4. O que significa otimizar a utilização do processador e restantes componentes do sistema?

Otimizar a utilização do processador e dos outros componentes do sistema significa usá-los da melhor forma possível. O processador deve estar ocupado a executar tarefas, e os outros componentes, como a memória e o disco, devem ser usados de forma eficiente para que o sistema funcione rápido e sem atrasos.

5. No Linux existem mecanismos de comunicação entre processos que são identificados pelos programadores por nomes/endereços pertencentes a diferentes espaços de nomes. Dê exemplos

- Pipes e Named Pipes (FIFOs)
- Sockets

- Semáforos
- Memória compartilhada
- Mensagens (Message Queues)

6. Considere o seguinte comando shell do Unix resultante do encadeamento de comandos elementares:

`ls -l | sort > saida`

- a) Descreva o resultado do comando.

O `"ls -l"` lista todos os ficheiros do diretório atual, incluindo informações detalhadas, o pipe `"|"` redireciona a saída do comando `"ls -l"` para a entrada do comando `"sort"`, que por sua vez, como o nome indica, ordena em os dados em ordem alfabética e redireciona `">"` para `"saida"`.

- b) Explique que mecanismos possibilitam este encadeamento de comandos.

São dois, os mecanismos que permitem o encadeamento de comandos, o *pipe* `"|"` que permite que a saída de um comando seja utilizada como entrada do comando seguinte e o `">"` que permite que o resultado de um comando seja guardado num ficheiro.

- c) Descreva sucintamente os principais passos do código que a shell executa para efetuar as operações deste comando encadeado.

A shell começa com a execução de `"ls -l"`, que gera uma listagem detalhada dos arquivos no diretório atual. Em seguida, a saída desse comando é passada como entrada para o comando `"sort"` através do operador *"pipe"* (`|`), que organiza as linhas da listagem. Por fim, a shell redireciona a saída ordenada para o arquivo `"saida"` com o operador `">"`, substituindo qualquer conteúdo anterior do arquivo. A shell gerência automaticamente o encadeamento entre os comandos e o redirecionamento da saída.