Trabalho Prático 4 – Sistemas Operativos

1. Explique o que entende por Processos em Sistemas Operativos

R: Em Sistemas Operativos, um processo é um programa em execução, gerido pelo sistema operativo. Inclui o código do programa, os dados, o estado atual e os recursos usados. O sistema operativo controla a criação, execução e terminação dos processos, garantindo que funcionem corretamente e em segurança.

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.

R: A memória partilhada permite que vários processos acedam diretamente à mesma área de memória, o que torna a comunicação muito rápida e eficiente. No entanto, exige que o programador implemente mecanismo de sincronização, como semáforos, para evitar conflitos no acesso simultâneo. Por outro lado, os objetos de comunicação do sistema, como pipes, filas de mensagem, facilitam a programação, uma vez que a sincronização é gerida pelo sistema operativo.

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

R: A memória partilhada é mais eficiente, pois permite a troca direta de dados entre processos, sem recorrer frequente ao sistema operativo, o que reduz a latência. Já os objetos de comunicação do sistema, como pipes ou filas de mensagem, são menos eficientes devido à sobrecarga de chamadas ao sistema e cópias de dados. No entanto, são mais fáceis de usar.

3. O que entende por Escalonamento?

R: Escalonamento é o processo através do qual o sistema operativo decide qual o próximo processo a utilizar o processador (CPU). Como há normalmente mais processos prontos a executar do que CPU's disponíveis, o escalonador escolhe gere a ordem e o tempo de execução de cada processo, garantido um uso eficiente dos recursos do sistema e tentando cumprir critérios como a justiça, desempenho e tempo de resposta.

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

R: Otimizar a utilização do processador e dos restantes componentes do sistema significa garantir que todos os recursos (CPU, memória, disco, etc.) são usados de forma eficiente, sem desperdícios nem tempos de espera desnecessários. Isto envolve manter o processador ocupado o máximo de tempo possível com tarefas úteis, minimizar o tempo de inatividade dos dispositivos, e equilibrar a carga entre os vários processos. O objetivo é melhorar o desempenho geral do sistema, reduzindo tempos de resposta e aumentando a produtividade.

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.

R: No Linux, os mecanismos de comunicação entre processos (IPC) podem ser identificados por nomes em diferentes espaços de nomes. Por exemplo, os sockets UNIX usam nomes no sistema de ficheiros como "/tmp/socket_exemplo". As filas de mensagens, a memória partilhada e os semáforos POSIX usam nomes iniciados por /, como "/minha_fila" ou "/meu_semaforo", cada um no seu próprio espaço de nomes. Estes nomes permitem aos processos localizar e utilizar os recursos de comunicação de forma organizada e sem conflitos.

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

   ls -l | sort > saída

   a) Descreva o resultado do comando.

R: O comando **ls -l | sort > saida** gera uma listagem detalhada dos ficheiros do diretório atual, ordena essa listagem e grava o resultado num ficheiro chamado ***saida***. Nada é apresentado no ecrã, pois a saída é redirecionada para esse ficheiro.

   b) Explique que mecanismos possibilitam este encadeamento de comandos.

R: O encadeamento de comandos em Unix/Linux é possível através de dois mecanismos principais: os **pipes** (|), que permitem enviar a saída de um comando como entrada para outro, e a **redireção** (>), que envia a saída de um comando para um ficheiro em vez do ecrã. Estes mecanismos são geridos pelo shell, que organiza a comunicação entre os comandos.

c) A shell executa os seguintes passos para processar o comando encadeado ls -l | sort > saida: cria dois processos, um para o ls -l e outro para o sort, e estabelece um pipe entre eles. A saída do ls -l é enviada para o sort, que ordena os dados. Em seguida, a shell redireciona a saída do sort para o ficheiro saida, substituindo o seu conteúdo. Por fim, a shell fecha os processos e os pipes.