

Kandayne Amabile Berto Borges

1º P - Ciência da Computação

Introdução a Ciência da Computação

## Lista C - Sistemas Operacionais e Compiladores

Exercício 1. Explique as seguintes funções dos seguintes componentes de um sistema operacional:

### • Gerenciamento de E/S

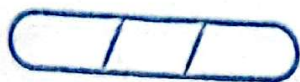
É responsável por gerenciar o acesso a dispositivos de E/S. Há limitações no número e na quantidade de dispositivos E/S em um computador, pois estes são mais lentos em comparação com a velocidade da CPU ou da RAM; quando um processo está acessando um dispositivo S/E, este fica indisponível para outros. O gerenciador de E/S, portanto, é responsável por usar eficientemente esses dispositivos. Isso acontece por meio de um monitoramento constante das atividades dos dispositivos E/S, da criação de filas e do controle de acesso. É importante destacar que a comunicação do gerenciador E/S do OS é feita por meio de drivers.

### • Gerenciador de memória

É responsável por alocar memória RAM. Os OS podem ser divididos em duas categorias de acordo com o gerenciamento de memória:







5

- Monoprogramação: aqui, a maior parte da capacidade de memória é dedicada a um único programa; o O.S é armazenado em uma pequena parte. Para que um programa seja executado, ele precisa estar inteiramente na RAM. Essa técnica traz os seguintes problemas:

- o programa só pode ser executado se couber na memória.
- se um programa pode ser executado por vez; com este programa esteja acessando algum dispositivo de E/S, que são mais lentos em comparação com a CPU. Isso é um desperdício de tempo.

- Multiprogramação: agora, múltiplos programas estão na memória ao mesmo tempo e são executados simultaneamente, com a CPU alternando rapidamente entre os programas. Existem dois tipos de multiprogramação: com ou sem swap.

### (1) Particionamento: (sem swap)

No particionamento, a memória é dividida em células de tamanho variado chamadas partições. Cada partição consegue executar um programa por vez, e um programador deve ser inteiramente instalado dentro da memória de uma única partição. A CPU percorre a memória executando os programas de maneira sequencial; cada programa é executado até que uma operação de E/S seja realizada. A CPU então salva o endereço



do programa na memória e executa o próximo. A paginação aumenta a eficiência da CPU, pois lida com o gargalo gerado pela diferença entre o tempo de processamento da CPU e RAM com os dispositivos de E/S, no entanto:

- o tamanho de cada partição é definido previamente: caso sejam pequenas demais, alguns programas não podem ser iniciados, e caso sejam grandes, sobra espaço vago na memória.
- com o uso do computador, esses "buracos" vazios não utilizados completamente podem ser criados. Reorganizar os programas de modo a resolver isso reduzirá o sistema.

## (2) Paginação (sem swap)

Agora, a memória é dividida em seções de tamanhos iguais, chamados quadros e programas em páginas. O tamanho de uma página e um quadro é usualmente a mesma. Cada página de um programa é armazenada em quadros de memória. A maior vantagem é que o programa pode ser armazenado de maneira não contínua, permitindo quadros em diferentes lugares. Isso aumenta a eficiência do uso da memória pois menos espaços vazios são gerados, porém, o programa ainda precisa estar inteiramente na memória para ser executado.

// continua no verso



### (3) Paginação sob-demanda (com swap)

Com processos simula a paginação clássica, em que programas são divididos em páginas e a memória em quadros de mesmo tamanho, a paginação sob demanda não requer que o programa seja inteiramente instanciado na memória RAM. O gerenciador de memória pode optar por inicializar o programa em partes, usando apenas as páginas que são necessárias naquele momento.

O restante do programa permanece na "memória virtual", algum dispositivo de memória secundária como SSD ou HDs. Essa solução permite que programas de tamanho maior que o da memória RAM sejam executados, em troca, o gerenciador de memória faz trocas (swap) entre as páginas que compõem um programa da memória virtual para a RAM.

### • Gerenciador de CPU

O gerenciador de processos gerencia a execução de programas pela CPU, alternando e determinando o estado de cada programa e formando filas de execução. Os programas podem assumir os seguintes estados:

- programas: é um conjunto de instruções armazenadas no disco.
- jobs: um programa se torna um job e passa do momento que entra para a fila de execução.



processo: é o programa sendo executado pela CPU ou esperando para ser executado. Enquanto que o job está na memória, se torna um processo.

Em outras palavras, para um programa ser executado, ele primeiro torna-se um job (entra na fila) e então um processo (é instanciado na memória). A alternância entre esses estados é determinada pelo gerenciador de processos. É usual que um programa alterne entre job e processo algumas vezes antes de ser executado.

Gerenciador de arquivos.

O gerenciador de arquivos controla o acesso, a criação, deleção, modificação, o armazenamento e a permissão de arquivos.

Exercício 2 O que são drivers de dispositivos e como eles se relacionam com o gerenciador de dispositivos? O que ocorre em um PC quando o OS não possui os respectivos drivers instalados? Os drivers de dispositivos formam uma camada de interface entre o gerenciador de dispositivos do EIS do sistema operacional e o hardware do computador. Essa interface permite que uma aplicação utilize os recursos do hardware, pois a aplicação conversa com o OS e este com o driver. Os drivers são, em geral, desenvolvidos pelos fabricantes e sabem como se comunicar com







0

aquele hardware. Sem o driver, o OS se torna incapaz de utilizar aquele recurso de hardware, o que pode levar a erros ou ao total não funcionamento de aquele recurso.

Exercício 3. Por que as Application Programming Interfaces dos OS formam um componente muito importante para os programadores? Explique.

APIs são um conjunto de funções, ou regras que permitem criar uma interface entre o S.O. e as aplicações, permitindo o acesso a funcionalidades do sistema operacional. APIs são importantes para programadores pois criam uma camada de abstração entre o OS e as aplicações.

Exercício 4. Qual é a importância de compiladores para os programadores? Explique.

Os compiladores criam uma ou mais camadas de abstração entre computadores e programadores. A "linguagem" assembly surge para permitir que programas possam ser escritos em texto, ao invés de linguagem de máquina (0's ou 1's). Porém, ainda era necessário que o programador tivesse o conhecimento completo daquela arquitetura de máquina que fosse programada. Com o tempo, programas deveriam ser escritos, isso ficou necessário trocar de computadores. Nesse contexto, surgem as linguagens de computadores e os compiladores. As linguagens de programação permitem que programadores escrevam seus programas mais rápido e outros possam

nos, tornando-as de mais fácil escrita e manutenção.  
Os compiladores então fazem a tradução  
de código fonte para o código de máquina ou  
para alguma linguagem intermediária. Mas de que  
modo um mesmo código de uma dada linguagem  
pode ser compilado para diferentes arquiteturas ou  
sistemas operacionais e depender do compilador  
usado.