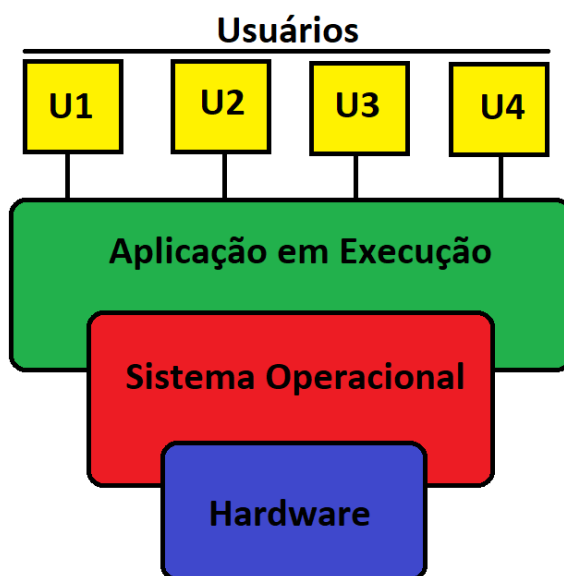


Nome:	WESLEY MARCOS BORGES			Curso:	GEC
Matrícula:	1651	Período:	P8	Matéria:	C012
				Turma:	

## Cap. 1 – Introdução aos Sistemas Operacionais

- 1) O sistema operacional é um middleware, ou seja, um intermediador entre o software e o hardware. Por fazer essa ponte, ele tem relação direta com ambos. Sua principal função além dessa ponte é facilitar a vida do usuário, disponibilizando recursos que agilizarão e tornarão possíveis as interações da forma que conhecemos. Algumas de suas funções são: controlar os dispositivos de I/O, fazer o controle de processos, fazer controle de memória e fazer controle de armazenamento. Segue uma representação gráfica de como ele pode ser visto:



- 2) As interrupções são ferramentas mais importantes de um SO, pois é através delas que ele controla quais aplicações serão executadas na ordem que hierárquica correspondente. Por exemplo, se é solicitada uma requisição de uma aplicação de usuário (com baixa prioridade) e está sendo executada uma aplicação de sistema (alta prioridade), o SO não atende à requisição da aplicação de usuário, devido a alta prioridade da aplicação de sistema. Existem dois tipos de interrupções, as de hardware e de software. As de **hardware** são de altíssima prioridade, pois são relacionadas aos periféricos usados, tendo de ser realizadas primeiro. Já as de **software** têm uma prioridade reduzida, já que programas enviam requisições à todo tempo (através do System Calls).
- 3) O Ciclo Fetch-Execute é uma operação de vital importante para que o SO consiga realizar um processamento. É composto por 3 partes, sendo:

**Fetch** = é feita uma busca na memória principal (RAM), que busca o dado na memória de armazenamento, e entrega o dado ao processador.

**Decode** = o dado coletado pela memória RAM enviado à CPU é decodificado, para que possa ser executado no próximo processo.

**Execute** = o dado coletado é executado pela CPU.

- 4) Esse processo de segurança é chamado Dual-Mode, que consiste em uma proteção oferecida pelo hardware através de um bit que indica se um código está sendo usado pelo SO. Esse processo funciona com 2 modalidades:

**Modo usuário** = esse processo é mais simplificado e contém muitas restrições para que o usuário apenas rode suas aplicações sem fazer alterações significantes. O bit de ativação é o bit 1.

**Modo kernel** = já o modo kernel é responsável por permitir ao usuário fazer alterações significantes na camada de aplicações do SO, podendo ser catastrófico, se não for usado com sabedoria. Seu bit de ativação é o bit 0.

- 5) Os controladores de dispositivos são hardwares responsáveis por manipular diretamente os dispositivos de I/O. São formados, geralmente, por buffers locais, possuindo a principal função de movimentar entre o buffer e os dispositivos que ele controla.
- 6) Essa comunicação é feita através de drivers, que são pacotes de instruções nas quais o SO se baseia para saber o que o dispositivo faz. Ele funciona como um manual. Alguns programas já vêm com drivers embutidos em sua ROM (firmware), geralmente os dispositivos plug-and-play, ou são baixados nos sites dos fabricantes.
- 7) Para operações de I/O demoradas, o SO utiliza uma técnica chamada DMA (Direct Memory Access), que consiste em uma transferência de um bloco inteiro de dados diretamente da memória para seu próprio buffer ou vice-versa. Dessa forma, apenas uma interrupção é gerada por bloco.
- 8) A inicialização de um SO ocorre através do bootstrap, um firmware que geralmente fica armazenado na ROM/EPROM e que inicia a BIOS (Basic Input Output System), que faz a verificação de todos os periféricos e hardwares. Após essa verificação são feitas algumas rotinas padrões e se tudo estiver em ordem, a BIOS carrega o kernel do SO, que passa a comandar todas as próximas operações.
- 9) O **processamento simétrico** utiliza um esquema onde todos os computadores estão no mesmo nível hierárquico, dessa forma os processos são divididos entre eles; é o SO (software) quem controla os processos. Já o **processamento assimétrico** utiliza um esquema Master-Slave, onde o processador Master é quem controla os processos e os distribui aos Slaves, ou seja, o controle é feito por hardware.

**10)** Os **sistemas multiprogramados** são aqueles que aumentam a utilização da CPU através de **Jobs**, que são um conjunto de tarefas que compõem uma unidade de trabalho. O SO mantém vários **Jobs** na memória principal, para que a CPU sempre tenha um deles para executar. Para que o SO consiga instruir a CPU à realizar os processos rápido o suficiente para parecer simultâneo, é necessário um esquema de chaveamento e encadeamento de processos, que é chamado de **Time-Sharing**. É uma extensão lógica da multiprogramação, que permite aos usuários com seus devidos processos compartilhem o computador em simultâneo.

**11)** As 4 gerências realizadas pelo SO são:

**Gerência de processos** = faz a gerência dos Jobs que estão sendo executados no momento.

**Gerência de memória** = faz a gerência da memória principal (RAM), alocando/desalocando memória quando necessário, definir a ordem de prioridade na execução e saber qual parte da memória está sendo usada e por quem.

**Gerência de armazenamento** = faz a gerência da memória secundária (HD/SSD), criando/apagando diretórios para organizar arquivos, mapeia arquivos para a memória secundária, criar/apagar arquivos; através do sistema de arquivos (NTFS, FAT32, etc).

**Gerência de I/O** = faz a gerência dos dispositivos de entrada e saída, que enviam dados ao SO.

**12)** Um **sistema operacional distribuído** trabalha através de clusters, que gerenciam um conjunto de processos e recursos, porém oferece a um usuário independente apenas algumas funções. Já um **sistema operacional de rede** funciona de forma local, podendo ser facilmente acoplado na rede, mas que atua independentemente.