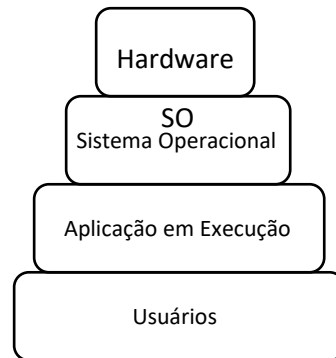


Nome: Pedro Gabriel Garcia Ribeiro Balestra	Matricula: 1551
Curso: GEC	Periodo: P8
	Matéria: C012

Cap.1 – Introdução aos Sistemas Operacionais

1. SO é o intermediador entre o software e o hardware (middleware), tendo uma relação direta entre ambos. E tem como objetivo facilitar a experiencia do usuário disponibilizando recursos como: controle de dispositivos de I/O, controle de processos, controle de memória e controle de armazenamento. Como podemos ver na representação gráfica:



2. São ferramentas que auxiliam o SO informando quais aplicações serão executadas seguindo a ordem de prioridade. As de maiores prioridades são as de Hardware, pois estão diretamente ligadas aos periféricos usados pelo SO, enquanto as de Software tem uma prioridade menor, como as requisições feitas por programas.
3. Fetch-Execute é uma operação importante para que o SO consiga realizar o processamento. Sendo composto por 3 partes:

Fetch -> É uma busca realizada na RAM (memória principal), pegando o dado na memória de armazenamento e entregando ao processador

Decode -> O dado coletado pela memória RAM enviado a CPU é decodificado, para que possa ser executado no próximo processo

Execute -> O dado é executado pela CPU

4. Utiliza o processo de segurança Dual-Mode, onde o hardware oferece uma proteção através de um bit que indica qual modo o SO está operando.

Modo Usuário -> Modo onde o usuário rode apenas suas aplicações sem que faça alterações significantes. Bit = 1

Modo Kernel -> Permite que o usuário realize alterações significativas na camada de aplicação do SO. Bit = 0
5. São Hardwares responsáveis por controlar dispositivos de I/O, compostos por buffers locais. Tem como função transitar entre os Buffers e os dispositivos que controla.
6. A comunicação é realizada através de drivers, são como manuais de instrução para o SO utilizar os novos dispositivos. Alguns já possuem os drives presentes na ROM, geralmente os dispositivos plug-and-play, os que não possuem podem ser baixados no site da fabricante

7. Para operação mais demoradas o SO utiliza o DMA, 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 ocorre através do bootstrap, que fica armazenado na ROM/EPROM e que inicia a BIOS, onde se faz toda a verificação dos periféricos e hardwares. Logo após a BIOS carrega o Kernel do SO que passa a comandar todas as próximas etapas.
9. **Processamento simétrico** -> utiliza de um esquema onde todos os computadores estão no mesmo nível hierárquico, dessa forma os processos são divididos entre eles, controle é feito pelo SO.
Processamento assimétrico -> utiliza de 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. Sistemas multiprogramados aumentam a utilização da CPU através de Jobs (Conjuntos de tarefas em uma unidade de trabalho). Para que a CPU consiga realizar processos rápidos, de maneira que pareçam simultâneos é necessário que haja o Time-Sharing (Chaveamento e Encadeamento de processos), processo que permite que os usuários compartilhem o computador em simultâneo.
11. **Gerência de processos** -> faz a gerência dos Jobs que estão sendo executados no momento.
Gerência de memória -> faz a gerência da memória principal, 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, criando/apagando diretórios para organizar arquivos, mapeia arquivos para a memória secundária, criar/apagar arquivos; através do sistema de arquivos.
Gerência de I/O -> faz a gerência dos dispositivos de entrada e saída, que enviam dados ao SO.
12. **Sistema operacional distribuído** -> trabalha através de clusters, que gerenciam um conjunto de processos e recursos, porém oferece a um usuário independente apenas algumas funções.
Sistema operacional de rede -> funciona de forma local, podendo ser facilmente acoplado na rede, mas que atua independentemente.