

Sistemas Operacionais

Notas de Aula

Francisco Rapchan
rapchan@writeme.com

Capítulo 1 - Introdução

- Apresentação do professor e leitura do cronograma.
- Objetivo do curso é entender o que é um sistema operacional e quais seus principais componentes.
- Exemplos de SO: MS-DOS, Windows, MacOS, UNIX, OS/2, MVS/VM,
- Porquê estudar os SO?

Introdução aos SO

Por mais complexo que possa parecer, um sistema operacional é nada mais que um conjunto de rotinas executadas pelo processador, da mesma forma que nossos programas. Sua principal função é controlar o funcionamento do computador como um gerente dos vários recursos disponíveis no sistema.

Um sistema de computação possui, normalmente, diversos componentes como terminais, impressoras, discos, fitas, etc. Quando utilizamos estes dispositivos não nos preocupamos com a maneira como é realizada esta comunicação e os inúmeros detalhes envolvidos. Uma operação aparentemente simples como ler um disquete exige um conjunto muito grande de rotinas específicas, como por exemplo, converter um endereço lógico em físico, posicionar a cabeça na trilha correta, esperar pelo setor correto passar, etc. O sistema operacional serve então como uma interface entre o usuário e os recursos do sistema, tornando esta comunicação transparente.

Em um sistema multiusuário, onde vários usuários podem estar compartilhando recursos tais como disco ou memória, é necessário que todos tenham chance de ter acesso a esses recursos de forma que um usuário não interfira no trabalho do outro. O sistema operacional é responsável por permitir o acesso concorrente a esses recursos de forma organizada e protegida, dando ao usuário a impressão de ser o único a utilizá-los.

Algumas definições de SO:

- É uma máquina virtual que faz com que a interface dos demais programas com o hardware seja mais simples.
- É a porção de software que roda no **modo kernel** (ou modo supervisor) com o objetivo de proteger o hardware da ação direta dos demais softwares. Estes outros softwares devem rodar no **modo usuário**.

- A função do sistema operacional é apresentar ao usuário uma máquina estendida ou uma máquina virtual equivalente ao hardware, porém muito mais simples de programar. Por exemplo, usar um comando de leitura de arquivos do disco tipo Read ("File") é conceitualmente muito mais simples do que se preocupar com coisas como o movimento das cabeças de leitura ou a velocidade dos discos do HD (embora para fazer uma leitura alguém tem que se preocupar com isto – que seja então o Sistema Operacional ao invés do programador de aplicativos).
- É um programa que controla todos os recursos do computador e fornece a base sobre a qual os programas aplicativos são escritos.
- A função do sistema operacional é gerenciar os usuários de cada um dos recursos da máquina.

Destas definições vemos que os SO podem ser apresentados como

- uma máquina virtual
- um gerente de recursos

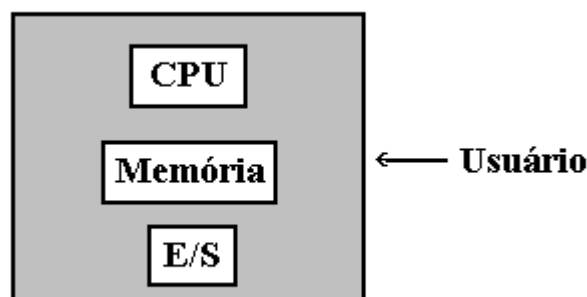
História dos SO

- Pedir para que leiam em cada o item 1.2 – História dos SO.
- Ver como Ken thompson encontrou um PDP-7 "abandonado" e escreveu para ele uma versão reduzida do sistema Multics; o UNIX.

Alguns conceitos básicos

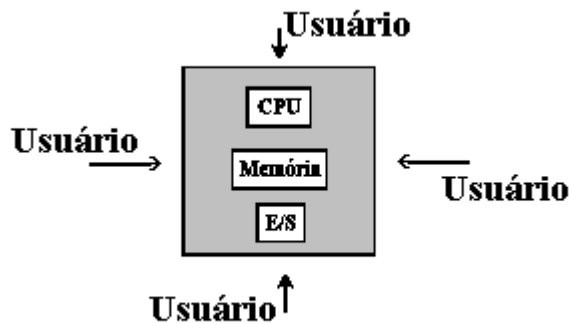
Monotarefa (single tasking).

- Este tipo de SO permite que apenas um aplicativo seja executado de cada vez.
- Os sistemas **monoprogramáveis** ou **monotarefa** se caracterizam por permitir que o processador, a memória e os periféricos fiquem dedicados a um único usuário. Nesses sistemas, enquanto o programa aguarda por um evento, como a digitação de um dado, o processador ficará ocioso, sem realizar qualquer tarefa útil. A memória é subutilizada caso o programa não a preencha totalmente, e os periféricos estão dedicados a um único usuário.
- Sistemas monoprogramáveis são de simples implementação, não tendo muita preocupação com problemas de proteção, pois só existe um usuário utilizando-o.



Multitarefa (multitasking).

- Podem executar várias tarefas ao mesmo tempo. Ex: imprimir um documento e fazer uma compilação.
- Os sistemas **multiprogramáveis** ou **multitarefa** são muito mais complexos e eficientes do que os sistemas monoprogramáveis. Nesses sistemas, vários usuários dividem os mesmos recursos, como memória, discos, impressoras, etc. Desta forma é possível aumentar a utilização do sistema, diminuindo assim, o custo total Máquina/homem.
- Num sistema multitarefa, enquanto um programa espera por uma operação de leitura ou escrita no disco, outros programas podem estar sendo processados no mesmo intervalo de tempo. Desta forma, a CPU estará sempre fazendo trabalho útil e teremos uma maior utilização da memória.
- Nos sistemas multitarefa os programas não rodam paralelamente. Na realidade, eles rodam seqüencialmente, um após o outro. A cada programa é dado uma fatia de tempo (**time-slice**) na qual ele executa uma parte. No fim deste tempo o sistema operacional suspende a execução deste programa e dá uma fatia de tempo a um outro programa. Esta fatia de tempo é muito pequena, de forma que a impressão que se tem é que todos os programas estão rodando ao mesmo tempo.
- Os sistemas multiprogramados podem ser classificados de acordo com a forma que interagem com os usuários, podendo ser divididos em sistemas **batch**, de **tempo compartilhado** e de **tempo real**.



Sistemas de Tempo Compartilhado

- Os sistemas de **tempo compartilhado** (*time-sharing*) permitem a interação dos usuários com o sistema, basicamente através de terminais de vídeo e teclado (interação *on-line*). Dessa forma, o usuário pode interagir em cada fase do desenvolvimento de suas aplicações e, se preciso, modificá-las imediatamente.
- Para cada usuário, o sistema operacional aloca uma fatia de tempo do processador. Nesses sistemas, não só o processador é compartilhado mas também a memória e os periféricos. O sistema cria para cada usuário um ambiente de trabalho próprio, dando a impressão de que todo o sistema está dedicado, exclusivamente, a ele.

Sistemas de Tempo Real

- Os sistemas de **tempo real** (*real-time*) são bem semelhantes em implementação aos sistemas de tempo compartilhado. A maior diferença é o tempo de resposta exigido na execução das tarefas. Enquanto em sistemas de tempo compartilhado o tempo de resposta pode variar sem

comprometer as aplicações em execução, nos sistemas de tempo real os tempos de resposta devem estar dentro de limites rígidos, que devem ser obedecidos, caso contrário, poderão ocorrer problemas irreparáveis.

- Nos sistemas de tempo real não existe a ideia de fatia de tempo, utilizada pelos sistemas time-sharing. Um programa executa o tempo que for necessário, ou até que apareça outro prioritário em função de sua importância no sistema. Esta importância ou prioridade de execução é controlada pela própria aplicação e não pelo sistema operacional.
- Esses sistemas são normalmente usados em controle de processos, como no monitoramento de refinarias de petróleo, controle de tráfego aéreo, usinas nucleares, etc.

Multiusuário.

- Permite que mais de um usuário trabalhe ao mesmo tempo usando os recursos de um computador.

Programas interativos.

- São os programas em que os usuários têm acesso via um terminal.

Programas batch ou em lote.

- Os sistemas **batch** (lote) caracterizam-se por terem seus programas armazenados em disco ou fita, onde esperam para ser executados sequencialmente. Quando um programa é iniciado ele tem a posse total do sistema até que sua execução termine. Quando um programa termina, um outro programa (também chamado **Job**) é colocado em execução imediatamente.
- Os programas que são executados em **batch** não precisam de interação com o usuário. Eles lêem e gravam seus dados de discos ou fitas.

Shell

- É o interpretador de comandos do UNIX.
- Acabou sendo o nome genérico de qualquer interpretador de comandos (ex: o command.com é o interpretador de comandos do MS-DOS).
- O shell não é o sistema operacional embora muitas pessoas pensem assim.

Interpretador de Comandos, também chamado de *shell*, é responsável pela interface conversacional com o usuário

Kernel

- É o núcleo do SO.
- Contém as rotinas mais críticas e mais frequentemente usadas.
- responsável pela gerência do processador, tratamento de interrupções e comunicação e sincronização de processos.

System calls

- Conjunto de instruções estendidas fornecidas pelo SO que criam, apagam e usam diversos objetos gerenciados pelo SO. Os mais importantes destes objetos são os processos e os arquivos.
- É através de system calls que os programas usuários solicitam serviços dos SO.
- São os "serviços" oferecidos pelo SO.

Trap

- Altera (ou "chaveia") a máquina do modo usuário para o modo kernel.
- A maioria dos processadores trabalha em dois modos: usuário (com limitações ao acesso ao hardware) e kernel (sem limitações de acesso).

Máquinas virtuais

- Máquinas virtuais. O principal exemplo é o VM/360. Estas máquinas são cópias fiéis do hardware, incluindo os modos kernel/usuário, entrada/saída, interrupções e tudo o mais que uma máquina real possui. Assim cada uma das VM pode rodar um sistema operacional diferente. [Ver Figura 1.11]

Spool

- Simultaneous Peripheral On Line.

Sistemas Operacionais são estudados dividindo-se suas principais funcionalidades. Tipicamente divide-se a funcionalidade de um sistema operacional em:

- Gerência de processos
- Gerência de memória: responsável pelo controle e alocação de memória aos processos ativos.
- Sistema de arquivo: responsável pelo acesso e integridade dos dados residentes em memória secundária (por exemplo disco, fita, etc).
- Controle de I/O (controle de entrada / saída): , responsável pelo controle e execução de operações de Entrada/Saída, bem como da otimização do uso dos periféricos.