

Sistemas Operativos, 2009

Licenciaturas em:

- Engenharia Informática
- Engenharia Electrotécnica

1. Introdução

1 O que é um sistema operativo?	2
1.1 Componentes dum Sistema de Computação	2
2 Evolução dos SO's	2
2.1 Os primeiros sistemas “bare machines”, antes dos anos 50 (1945–55) Vacuum Tubes and Plugboards	3
2.2 Sistemas Monoprogramados de tratamento por lotes (Batch Systems) - 1º Sistema Operativo, anos 50, desenvolvido pela General Motors para um IBM701.....	3
2.3 Sistemas Multiprogramados de tratamento por lotes (Multiprogrammed Batch Systems) - anos 50	4
2.4 Sistemas Multiprogramados Interactivos de partilha de tempo	4
2.5 Sistemas de Tempo Real - anos 60.....	4
2.6 Sistemas Paralelos - anos 60	4
2.7 Sistemas Distribuídos - anos 80	4
2.8 Sistemas Embutidos e diapositivos móveis. (Very Small Computers).....	5
4 Principais Tarefas dum Sistema Operativo	8
5 Componentes dum Sistema Operativo	9
Spooling	10

1 O que é um sistema operativo?

O software de qualquer computador ou sistema computacional pode ser dividido basicamente em dois tipos: *Software do sistema* - programas que contribuem para o controlo e desempenho das operações do computador e *Software de aplicação* - programas que resolvem os problemas dos usuários

O mais fundamental dos programas do sistema é o *sistema operativo*. É este que controla todos os recursos do computador e é este que fornece a base sobre a qual os programas de aplicação podem ser escritos.

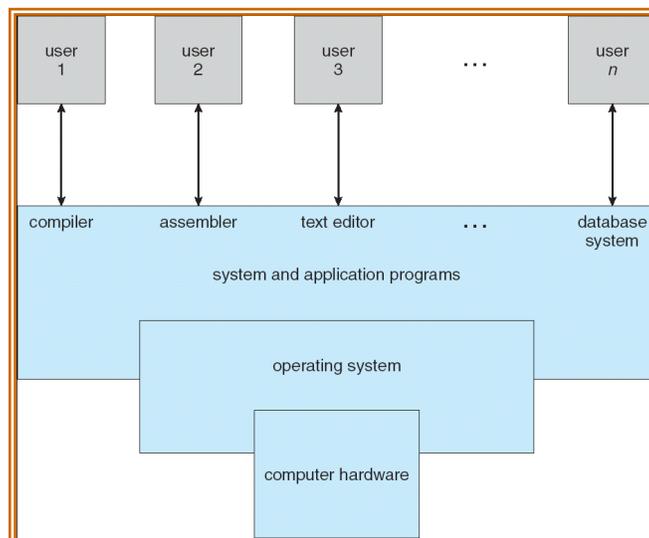
Definição preliminar dum Sistema Operativo:

É um programa ou até um conjunto de programas que age como intermediário entre o utilizador e o hardware ou máquina física. Os objectivos dum SO são:

- Executar programas do utilizador e tornar mais fácil a resolução de problemas.
- Tornar fácil o uso da máquina e integre todos os components dum sistema de computação
- Utilizar o hardware do computador duma forma eficiente.

1.1 Componentes dum Sistema de Computação

- Hardware – os recursos físicos: CPU, memória, dispositivos de Entrada e Saida (E/S).
- Sistema operativo – controla e coordena a utilização do hardware durante a execução de vários programas de aplicação, eventualmente pertencentes a diferentes utilizadores.
- Programas de aplicação – definem as formas de utilização dos recursos do sistema com o objectivo de resolver problemas dos utilizadores (compiladores, sistemas de bases de dados, jogos de vídeo, programas de contabilidade, etc.).
- Utilitários – Conjunto de programas de sistema que têm funcionalidades úteis.
- Utilizadores -pessoas, máquinas ou até outros computadores. O utilizador final do sistema que não está (ou não quer estar) preocupado com a sua arquitectura.



2 Evolução dos SO's

2.1 Os primeiros sistemas “bare machines”, antes dos anos 50 (1945–55) Vacuum Tubes and Plugboards

Nas primeiras máquina as mesmas pessoas, desenhavam, projectavam, fabricavam e programavam e operavam as máquinas. Aqui o utilizador era também programador e operador do sistema Toda a programação foi feito numa linguagem maquina e muitas vezes necessitava um “rewiring” do próprio computador. Estas maquinas não dispunham de SO, apenas tinham de um monitor de controlo. Este permitia carregar os programas em memória, utilizando primeira “plugboards” e depois as cartões perfurados, e fazer pequenas mudanças e verificar a execução do programa Nestes sistemas verificava-se o uso ineficiente dos recursos – por exemplo quando era preciso mudar uma fita ou toner na impressora a maquina estava parada.

2.2 Sistemas Monoprogramados de tratamento por lotes (Batch Systems) - 1º Sistema Operativo, anos 50, desenvolvido pela General Motors para um IBM701.

Nestes sistemas as tarefas, sem interacção com o utilizador, eram organizadas em grupos (ou lotes) e eram submetidas ao sistema para execução. Um lote consistia dum conjunto de tarefas (jobs) cada tarefa consistia dum programa e todos os dados do input necessários.

Os lotes foram por exemplo organizados por algum criterio, por exemplo um lote onde todos as tarefas precisavam o mesmo conjunto de “fitas/tapes”, ou. “Um grupo que demorava uma hora de tempo de cpu”. Sempre que uma tarefa se encontrava em execução esta tinha controlo total sobre o computador. Terminada a tarefa o controlo era devolvido ao SO que "limpava" o sistema e iniciava a próxima tarefa. Portanto uma Característica Principal dos *Sistemas Batch* –é a falta de interacção entre utilizador e a sua tarefa (job), Uma vez submetida para execução uma tarefa de batch pode ficar horas a espera de ser executado dependente da carga no sistema.

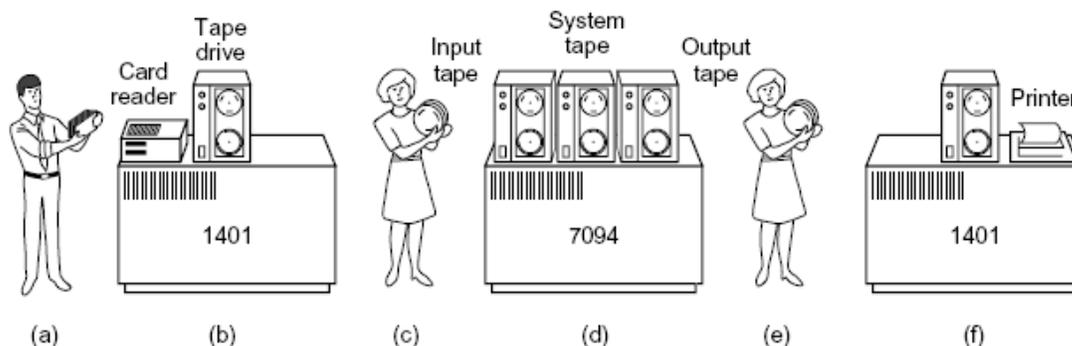


Figura (Livro do Tannebaum) Um sistema batch. (a) Programadores trazem os cartões do programa para a máquina IBM 1401. (b) 1401 lê cartões criando uma sequência de tarefas (batch) numa fita. (c) O operador leva as fitas para a IBM 7094, uma máquina mais poderosa e mais cara. (d) 7094 efectua a computação. (e) Operador leva a fita do output para a 1401. (f) 1401 imprime o output

2.3 Sistemas Multiprogramados de tratamento por lotes (Multiprogrammed Batch Systems) - anos 50

Devido à lentidão dos dispositivos de leitura, passou a ser carregada mais do que uma tarefa em memória, permitindo reduzir os tempos mortos através da alternância (multiplexing) entre tarefas. A execução concorrente de vários programas é dado o nome de multiprogramação. Note que isto é diferente do multiprocessamento que descreve a situação onde mais dum CPU é presente num sistema computacional permitindo a execução simultanea de duas tarefas

A Necessidade duma maneira de escolher a tarefa e controlar a troca das tarefas (SO)

2.4 Sistemas Multiprogramados Interactivos de partilha de tempo

(Time-Sharing Systems) - CTSS, 1961, MIT

Complementando os sistemas multiprogramados, passou a ser possível ter mais do que um utilizador a trabalhar simultaneamente no mesmo computador, em diferentes terminais, realizando o sistema uma multiplexagem entre os vários utilizadores. Este utilizador passou a ter acesso constante à máquina.

→ Necessidade de SO escolher a tarefa e controlar a troca – agora mais complicado
Utilizadores “online” precisavam de ter e aceder e a um sistema de ficheiros – conceito abstracto gerido pelo SO.

2.5 Sistemas de Tempo Real - anos 60

Estes surgiram para controlar processos industriais e dispositivos militares. Os sistemas de tempo real caracterizam-se por obedecerem a restrições de tempo para a execução de determinadas tarefas.

2.6 Sistemas Paralelos - anos 60

Estes surgiram como sequencia natural da tentativa de adicionar mais recursos ao sistema nomeadamente mais processadores - duas ou mais CPU's.

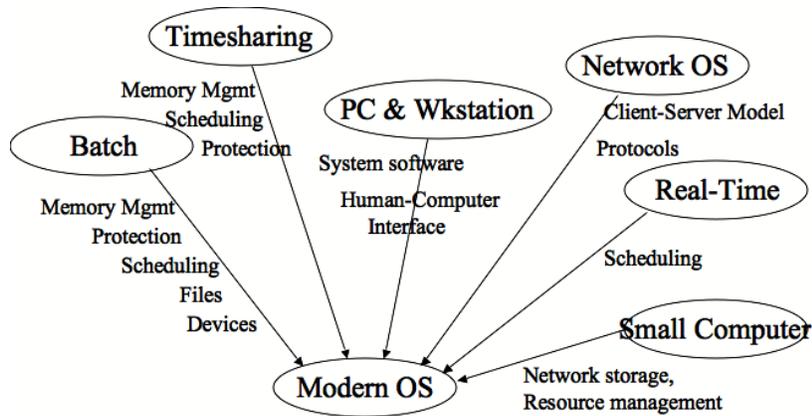
2.7 Sistemas Distribuídos - anos 80

Os SOs distribuídos apareceram na sequência do desenvolvimento dos mecanismos de interligação entre computadores. A carga computacional é distribuída por vários computadores.

2.8 Sistemas Embutidos e dispositivos móveis. (Very Small Computers)

PDAs, STBs, Telemóveis, Touch Screen Devices etc.

Um Novo Class de Sistemas Operativos como : PalmOS, PocketPC(WinCE), Symbian, Embedded Linux, Android etc



Description of the IBM 701

<http://www.palosverdes.com/lasthurrah/>

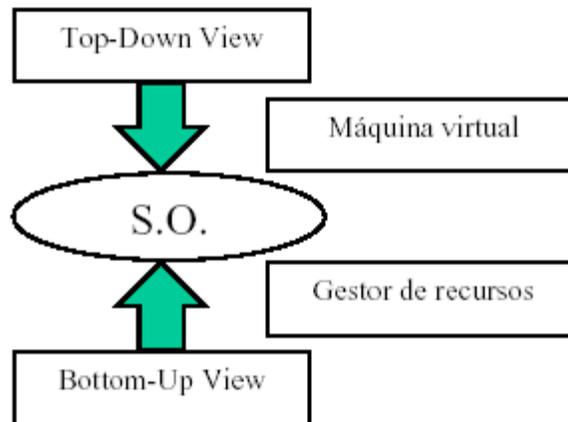
Among the outstanding features of the 701 are its large-capacity high-speed electrostatic storage, intermediate magnetic drum storage, magnetic tape units, a versatile and fast input-output system, and computing speed characterized by a multiplication time of 456 microseconds.

In order to achieve maximum versatility, every function of the machine is under control of the stored program. This versatility allows the machine to execute instructions at the rate of about 14,000 per second on typical problems. Also functions such as input-output operation, which are determined by fixed circuitry on some computers, are under complete control of the program and, hence, under complete control of the operator. The great advantage of this system lies in the fact that a customer may build up a library programs which will accomplish his special applications at peak machine efficiency. No compromise in efficiency is necessary in the design of the machine to accommodate an average application. Furthermore, a customer may efficiently calculate on any 701 installation simply by using his own library of programs."

Each run was limited to 10 minutes so only short jobs, usually subroutines could be checked out in this manner. The incident I remember most was working with an older operator who had a degree in mathematics. At that time, some clown thought that you had to have a college degree to operate the "Giant Brain." It was a great step forward when they realized that the operators of the IBM card equipment made the best operators. The mathematicians who operated the 701 wouldn't stay long due to boredom and the ones that stayed were too dumb to learn how to run the computer. Well, the operator I am talking about is in the later category. I was checking out a program during Happy Hour and was watching this mathematician put my cards in the card reader. My program stopped and he turned to me and inform me the computer had stopped. Before I had a chance to record the stop address, he had cleared the console. A week later, the powers that be realized he was never going to become an even semi-competent operator. Of course nay of us programmers could give him instructions on what to do, but that wasn't allowed. After the card operators began to run the computer, things got a lot better.

3 Motivos históricos para o desenvolvimento do SO

Há muito tempo tem-se tentado libertar o usuário do computador da complexidade do hardware. Isto tem sido feito colocando-se uma camada de software sobre o hardware de modo que este software gere todas as partes do sistema. Assim o usuário seria apresentado com uma interface que é mais fácil de entender e de programar. Esta interface é chamada de máquina virtual é uma vista do sistema do Cima para baixo (Top-Down). A vista alternativa dum sistema operativo, do nível do Hardware, é uma vista “Bottom Up” e visualize o sistema operativo com um gestor do próprio hardware. Estas duas vistas são visualizadas na figura em baixo:



SO como Máquina Virtual (Top-Down View)

Transformar um conjunto diversificado de circuitos electrónicos, discos e periféricos numa máquina simples de utilizar.

- O Utilizador não pretende envolver-se demasiado nos detalhes do Hardware
- SO virtualiza o H/W. A abstracção apresentada ao utilizador pretende-se simples e de fácil compreensão
- Apresenta ao utilizador uma interface que trata de um modo uniforme operações sobre entidades semelhantes
- Garantir fiabilidade e segurança

Sistemas Operativos também fornecem recursos virtuais (ficheiros, base de dados, interpretação de multi-media etc.). Muitos vezes caumentam a funcionalidade basico da maquina com memoria virtual, processamento concorrente e multiplos linhas de controlo (threads). Historicamente OS's tem oferecido cada vez mais recursos

SO como Gestor de Recursos (Bottom-up view)

Obter o máximo rendimento do *hardware* através da sua utilização para o processamento de um grande conjunto de actividades.

- Garantir uma gestão dos recursos
- Optimização do desempenho

4 Principais Tarefas dum Sistema Operativo

Execução de Programas: É necessário realizar um conjunto de tarefas para a execução de um programa. Instruções e dados têm de ser carregados em memória, dispositivos de I/O(Entrada/Saída) e ficheiros têm de ser inicializados. Mecanismos de partilha prontos.

Acesso a dispositivos de Entrada/Saída: Cada dispositivo de I/O tem o seu conjunto de instruções, sinais de controlo e operação. O SO torna transparentes as particularidades dos vários periféricos.

Acesso controlado a ficheiros: No caso dos ficheiros, para além da natureza do dispositivo de I/O, o SO deve permitir ler ficheiros em vários formatos e protege-los de acordo com o tipo de acesso permitido aos vários utilizadores.

Criação de Programas: O SO disponibiliza facilidades e serviços, como editores e debuggers, que se encontram na forma de utilitários que não fazem verdadeiramente parte do SO.

Acesso ao sistema: No caso de um sistema partilhado, deve ser controlado o acesso ao sistema como um todo e especificamente a cada recurso.

Deteção de erros e resposta: O sistema deve dar resposta a uma grande variedade de erros que podem acontecer tanto no hardware como no software.

Registo de parâmetros: Um bom SO permite registar estatísticas relativas a vários recursos e monitorizar parâmetros de desempenho. Esta informação permite o ajuste do sistema aumentando o seu desempenho.

5 Componentes dum Sistema Operativo

Gestão de Processos: Sendo um *processo* um programa em execução, devem ser disponibilizadas sobre eles as seguintes funções básicas: criação, eliminação, suspensão, activação, sincronização e comunicação. Adicionalmente o SO deve garantir uma partilha correcta do tempo de CPU entre os vários processos.

Gestão de Memória: Sendo a memória o dispositivo onde são carregados os processos em execução e o próprio SO, este deve permitir a reserva, libertação e monitorização do espaço em memória. Se o SO dispuser de um mecanismo de memória virtual então o espaço disponível para os processos poderá exceder a memória física (RAM) disponível no sistema.

Gestão de I/O(Entrada/Saída): Um sistema computacional dispõe de vários periféricos sobre os quais são possíveis operações de leitura (entrada) e/ou de escrita (saída). Então para cada dispositivo (ou grupos de dispositivos) o SO dispõe de módulos que lidam especificamente com cada tipo de dispositivos (device drivers).

Gestão de Ficheiros: Sendo um *ficheiro* uma colecção de informação relacionada, devem ser disponibilizadas sobre eles as seguintes funções básicas: criação, eliminação e manipulação. Adicionalmente deve existir a possibilidade de organizar os ficheiros (em directorias) e de realizar cópias de segurança (backups).

Protecção: O SO poderá disponibilizar mecanismos de protecção para controlar os acessos dos processos aos recursos e dos utilizadores ao sistema.

Interpretador de comandos: Muitos dos comandos passados ao SO são instruções de controlo que lidam com os mecanismos de gestão do SO. O programa que lê e interpreta essas instruções de controlo é o interpretador de comandos.

Chamadas de sistema: A comunicação entre os processos e o SO é realizada através de chamadas de sistema (system calls). Estas são funções disponibilizadas através de bibliotecas (APIs-Application Programming Interfaces) pelo SO e podem ser utilizadas em qualquer programa. A chamadas do sistema criam, apagam e utilizam vários objectos de software e estruturas de dados geridos pelo S.O. Os mais importantes destes objectos são os *processos* e os *arquivos*.

6 Nota acerca do Spooling

O **spooling** refere-se ao processo de colocar tarefas em um **buffer**, uma área especial na memória ou no disco, onde um dispositivo pode acessá-las quando estiver pronto. Isso é semelhante a um carretel de linha em uma máquina de costura: a pessoa coloca a linha no carretel, e a máquina a puxa conforme necessário.

O **spooling** é útil porque os dispositivos acessam dados em velocidades diferentes. O **buffer** atua como uma estação de espera onde os dados permanecem enquanto o dispositivo mais lento processa as informações. Os dados são adicionados e removidos apenas nas extremidades do buffer, sem acesso aleatório ou edição. Isso também permite que a CPU execute outras tarefas enquanto aguarda o dispositivo mais lento concluir sua operação.

A aplicação mais comum de **spooling** hoje é o **spooling de impressão**. Nesse caso, os documentos são carregados em um **buffer** (geralmente uma área no disco), e a impressora os processa conforme sua própria velocidade. Isso permite que o usuário continue realizando outras tarefas no computador enquanto a impressão ocorre em segundo plano. Além disso, o **spooling** permite que vários documentos sejam colocados em uma fila de impressão, em vez de exigir que cada impressão seja concluída antes de iniciar a próxima.

Origem do termo

Fita magnética de gravação enrolada em um carretel ou bobina.

A palavra "**spool**" supostamente é um acrônimo para "**simultaneous peripheral operations on-line**" (operações periféricas simultâneas online).

Nos primeiros computadores mainframe, os discos rígidos eram relativamente pequenos e muito caros em comparação com os padrões atuais. Devido a esse alto custo, era necessário reservar os discos para arquivos que exigiam acesso aleatório, enquanto arquivos grandes e sequenciais eram gravados em bobinas de fita magnética.

Programas típicos rodavam por horas e produziam centenas ou milhares de páginas de saída impressa. Periodicamente, o programa interrompia a impressão para realizar buscas ou ordenações demoradas. No entanto, era desejável manter as impressoras funcionando continuamente. Assim, enquanto o programa estava em execução, ele gravava o arquivo de impressão em uma fita **spool**, que posteriormente era lida pelo programa responsável pelo controle da impressora.