

Sistemas Operacionais

Introdução

Informações Gerais

- Site
 - <http://www.inf.ufes.br/~rgomes/so.htm>
- Email
 - soufes@gmail.com

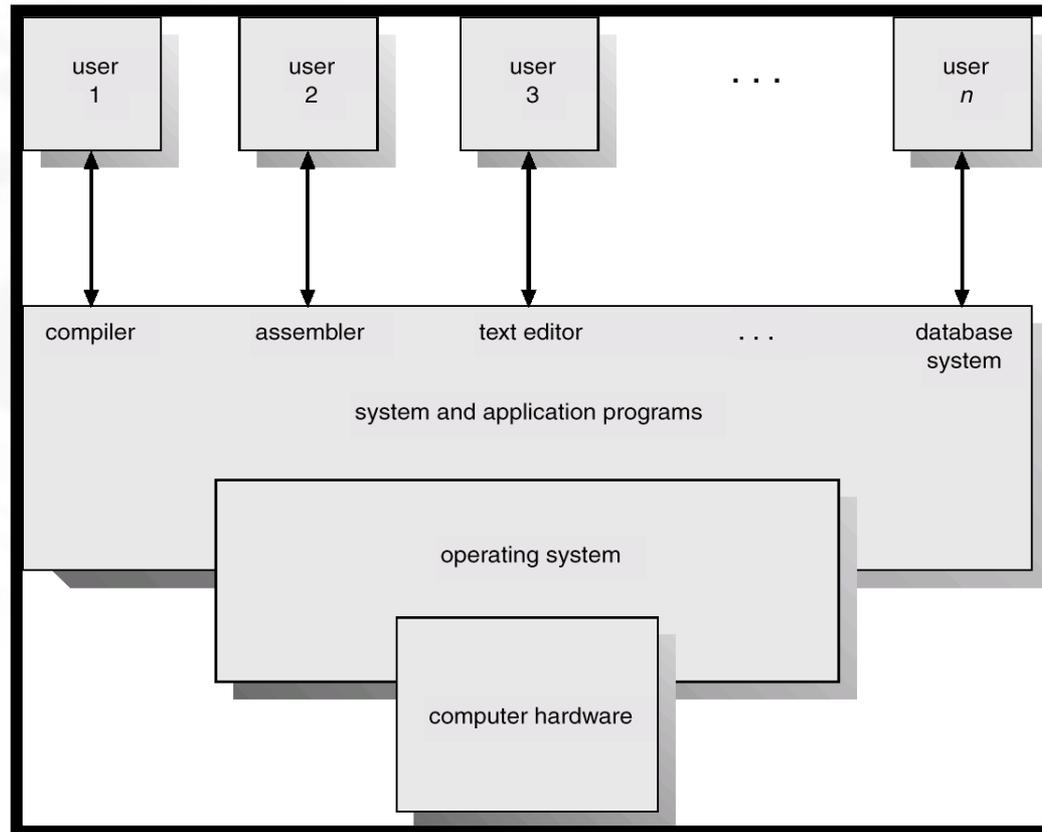
Objetivo do Curso

- Apresentar os fundamentos teóricos dos sistemas operacionais modernos, enfatizando os seus aspectos...
 - de organização interna (arquitetura conceitual)
 - de estruturas e mecanismos de implementação.

Sistema de Computação

- Hardware
 - Provê os recursos básicos de computação (UCP, memória, dispositivos de E/S).
- Programas de aplicação
 - Definem as maneiras pelas quais os recursos do sistema são usados para resolver os problemas computacionais dos usuários (compiladores, sistemas de banco de dados, video games, programas financeiros, etc.).
- Usuários
 - Pessoas, máquinas, outros computadores.

Visão Abstrata (1)



Um Sistema Operacional...

- ... possibilita o uso **eficiente** e **controlado** dos diversos componentes de hardware do computador (unidade central de processamento, memória, dispositivos de entrada e saída).
- ... implementa políticas e estruturas de software de modo a assegurar um melhor **desempenho do sistema** de computação como um todo.

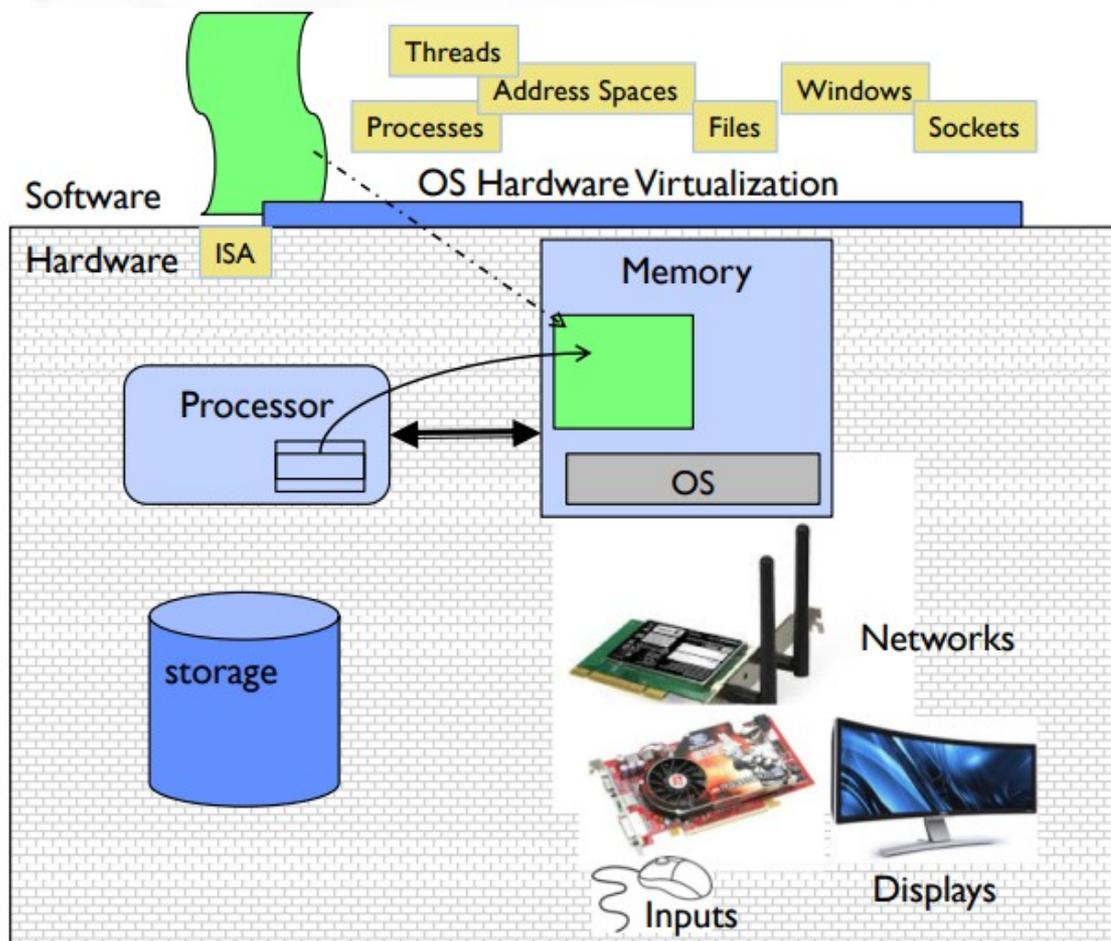
Definição

- Nome dado a um conjunto de programas que trabalham de modo cooperativo com o objetivo de prover uma máquina mais **flexível** e **adequada** ao programador do que aquela apresentada pelo hardware sozinho.
 - **Interface de programação**
 - **Gerenciamento de recursos**

Função

- “A principal função de um sistema operacional é prover um ambiente no qual os programas dos usuários (aplicações) possam rodar.
- Isso envolve:
 - Definir um **framework básico** para a execução dos programas
 - Prover **abstrações** para as aplicações
 - File systems, Processes, threads, VM, containers, Naming system, etc.
 - Oferecer uma **interface de programação** para acesso aos recursos

Organização Típica

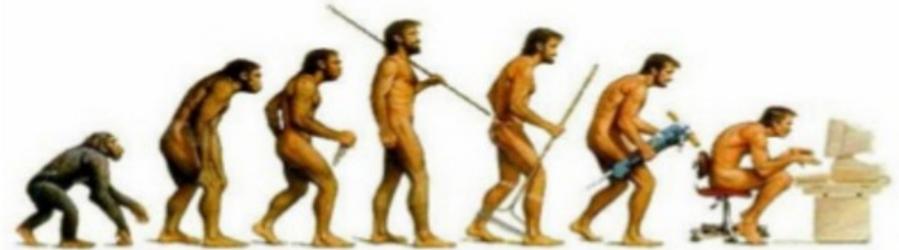


Organização Típica

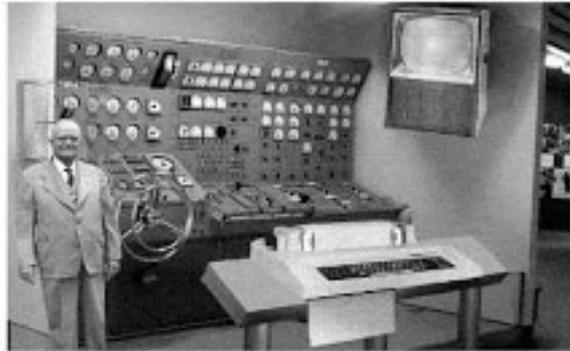
- **Núcleo (kernel)**
 - Responsável pela gerência do processador, tratamento de interrupções, comunicação e sincronização entre processos.
- **Gerente de Memória**
 - Responsável pelo controle e alocação de memória aos processos ativos.
- **Sistema de Arquivos**
 - Responsável pelo acesso e integridade dos dados residentes na memória secundária.
- **Sistema de E/S**
 - Responsável pelo controle e execução de operações de E/S e otimização do uso dos periféricos.

- **Processador de Comandos (shell) / Interface com o Usuário**
 - Responsável pela interface conversacional com o usuário.

Uma Breve História



- No início ... **inexistência de S.O.!**
 - O usuário é programador e operador da máquina ao mesmo tempo.
 - *Setup* incluía carregar o compilador e o programa fonte, salvar o programa compilado, carregar e link-editar o programa executável.
- O sistema operacional surge com o objetivo básico de **automatizar** a preparação, carga e execução de programas.
- Também torna-se fundamental para otimizar o uso dos recursos de hardware pelos programas.



First Generation



Second Generation



Third Generation



Fourth Generation



Fifth Generation

1a Geração (década de 50)	Execução automática de jobs JCL – Job Control Language Ex: IAS , ATLAS , IBM 701	HW de 2a geração, com circuitos transistorizados
2a. Geração (início da década de 60)	Primeiros sistemas de multiprogramação e hardware com multiprocessamento. Sistemas de tempo real . Ex: IBM OS/MTF , CTSS (IBM 7094) , MCP (Burroughs 5000, etc.) .	HW de 3a. geração, com circuitos integrados..
3a Geração (meados dos anos 60 a meados dos anos 70)	Sistemas multi-modo e de propósitos gerais. Uso de memória virtual . Sistemas complexos e caros, à exceção do UNIX. Primeiros computadores pessoais. Ex: Multics (GE 645) , TSS (IBM 360/67) , CDC Kronos (CDC 6000) , RCA VMOS, etc.	HW construído com tecnologia VLSI
4a. Geração (meados dos anos 70 e início dos anos 80)	Grandes sistemas de multiprocessamento , uso intensivo de teleprocessamento , sistemas de memória virtual. Início da comercialização PCs Ex: MCP (Burroughs B6700) , VMS (VAX 11/760) , MVS (IBM 370) , etc.	HW com suporte de microprogramação
5a. Geração (Anos 80 e 90)	Arquiteturas distribuídas , ambiente de redes de computadores, máquinas virtuais , uso intenso de microcomputadores pessoais, interfaces visuais mais elaboradas. (Ex: DOS , MAC OS , Windows , Unix-like OS , IBM OS/2 , IBM VM/370)	Grande diversidade de HW de E/S, UCP e memórias de alta velocidade. Arquiteturas RISC
Dias atuais	Arquitetura microkernel e multithreading, sistemas multiplataforma, middleware, projeto orientado a objetos,, suporte a computação móvel, etc. Ex: Linux , Windows2000... 7, 8, Palm OS , IOS , Android , Solaris , Unix SVR4 (... AIX) , Linux embarcado , VMs , Containers , etc.	HW para multiprocessamento simétrico, HW para computação móvel e ubíqua, sistemas embarcados, IoT

Tipos de Sistemas Operacionais

- **Sistemas Mainframes**
- **Sistemas Desktop**
- **Sistemas Handheld (mobile devices)**
- **Sistemas embarcados**

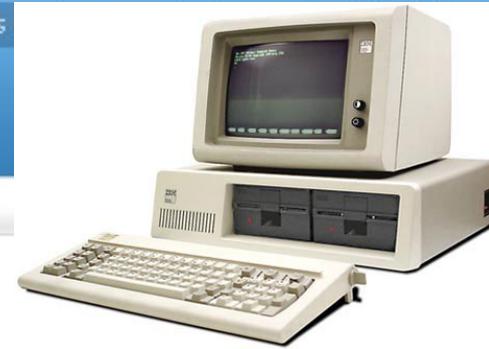
- Sistemas de Lotes (Batch)
- Sistemas de Tempo Compartilhado (Time Sharing)
- Sistemas de Tempo Real (Real-Time)

Sistemas *Mainframes*



- Primeiros sistemas computacionais usados na resolução de problemas comerciais e aplicações científicas.
- Nos dias atuais, são projetados para processamento de grande volume de dados.
- Arquitetura de hardware complexa.
- Trabalham em multimodo (usualmente “batch” e “time-sharing”).





Sistemas *Desktop*

- Caracterizado pelo uso de computadores pessoais.
- Diferentemente dos sistemas *mainframes*, é um sistema de computação geralmente dedicado a um único usuário.
- Dispositivos típicos de E/S: teclado, mouse, terminal de vídeo, pequenas impressoras.
- Apresenta conveniência para o usuário, com um bom tempo de resposta e uma boa taxa de processamento a um custo relativamente baixo.
- Pode rodar diferentes tipos de S.O.
- (Windows, MacOS, Linux, etc.)





Sistemas *Handheld*

- Sistemas operacionais adaptados para dispositivos móveis, como Smartphones e Tablets...
- No **início**...memória limitada, processador mais lento e display de pequenas dimensões.
- S.O. e aplicações projetados para minimizar o uso do processador (redução do consumo da bateria).
- Uso possível de **tecnologias wireless**, como Bluetooth e wi-fi, para acesso remoto a outros dispositivos e internet.
- Grande variedade de acessórios que expandem a sua funcionalidade.

Sistemas Embarcados

- Limitações ainda maiores de memória, processamento, consumo de energia e confiabilidade
- Ex de OS para IoT
 - ├ TinyOS, FreeRTOS, Contiki



- Sistemas operacionais reprojitados...



Tipos de Sistemas Operacionais

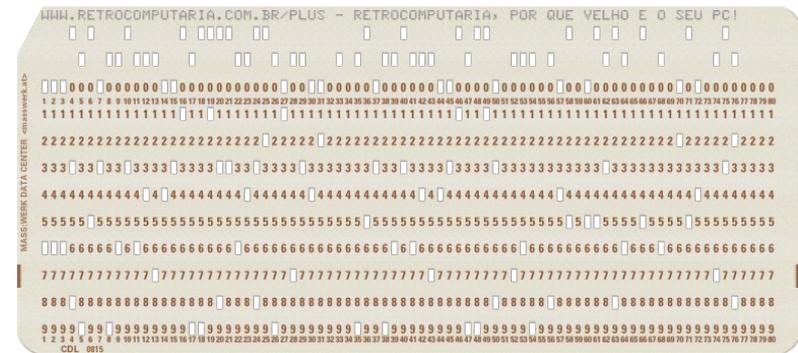
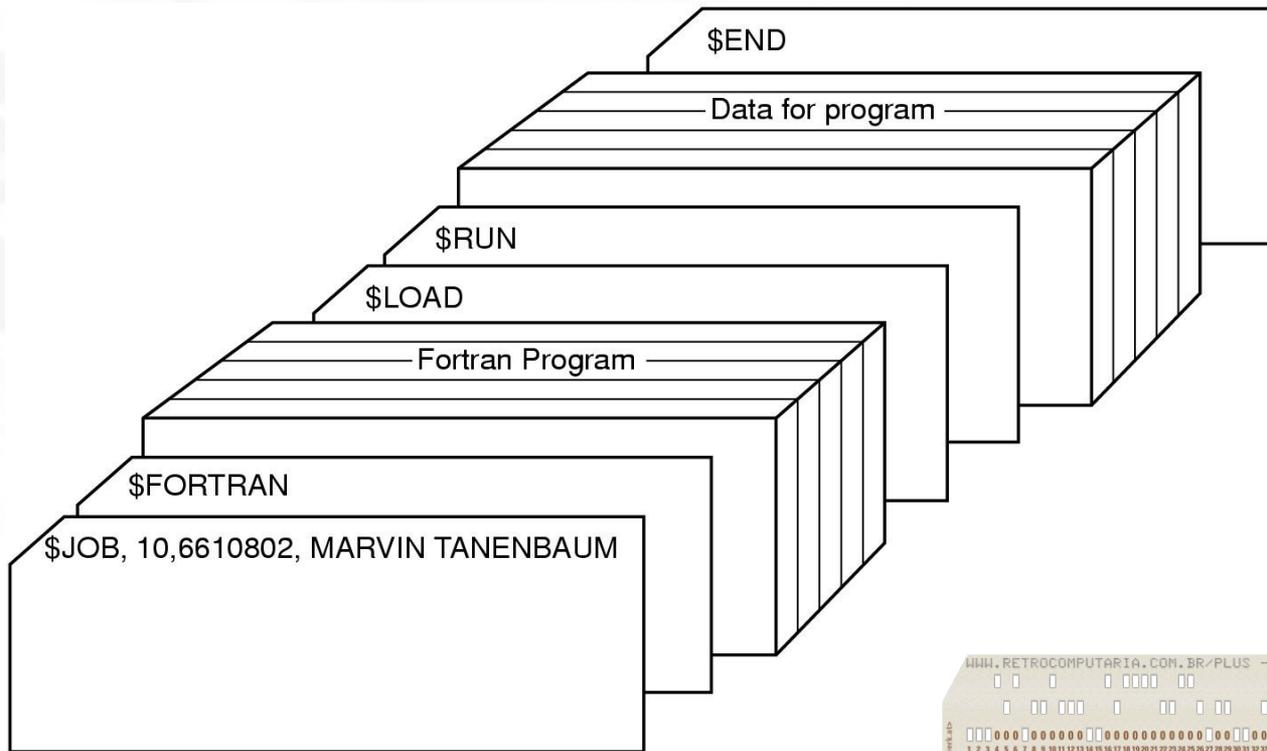
- Sistemas Mainframes
- Sistemas Desktop
- Sistemas Handheld (mobile devices)
- Sistemas embarcados

- **Sistemas de Lotes (Batch)**
- **Sistemas de Tempo Compartilhado (Time Sharing)**
- **Sistemas de Tempo Real (Real-Time)**

Sistemas de Lotes (1)

- Primeiro tipo de sistema operacional
- *Job* é o conceito básico: constituído do programa a ser compilado e executado, acompanhado dos dados.
 - *Jobs* semelhantes são organizados em lotes, de modo a reduzir o tempo de *setup*.
- Sequenciamento automático de *jobs*, com transferência automática de controle de um *job* para outro (processamento seqüencial).
 - Inexistência de computação interativa.
- Número de tarefas processadas por unidade de tempo é alta devido à seqüencialidade, sem interrupção.
- Tempo **médio de resposta** pode ser muito alto (ex: presença de *jobs* grandes à frente de *jobs* pequenos).

Sistemas de Lotes (2)



Sistemas de Lotes (3)

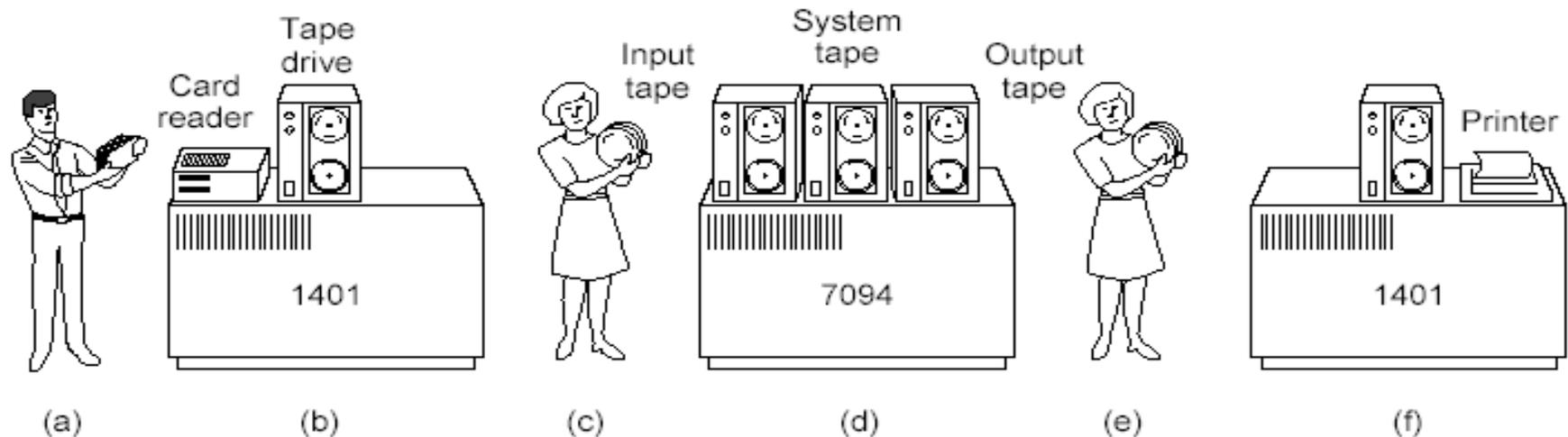
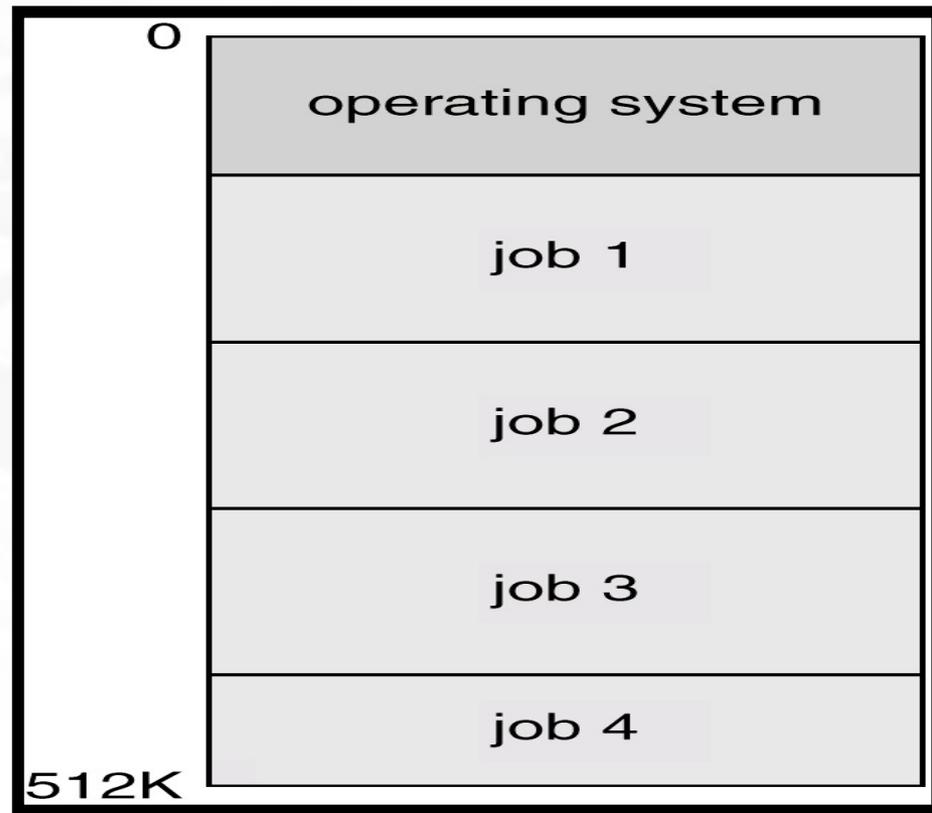


Figure 1-2. An early batch system. (a) Programmers bring cards to 1401. (b) 1401 reads batch of jobs onto tape. (c) Operator carries input tape to 7094. (d) 7094 does computing. (e) Operator carries output tape to 1401. (f) 1401 prints output.

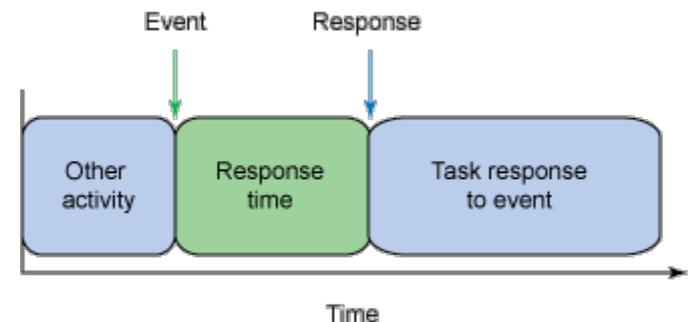
Layout da MP em Sistemas de Lotes



Há **multiprogramação**, porém sem tempo compartilhado.
(*Batch Multiprogramming*)

Sistemas de Tempo Compartilhado

- A capacidade e o tempo de processamento da máquina é dividida entre múltiplos usuários
- Antigamente, o acesso era por meio de **terminais “burros”**
 - Comandos de usuários são interpretados e executados *on-line*.
- Uso da técnica de **multiprogramming + time sharing** para a manipulação de múltiplos *jobs* interativos.
 - *Surgiram os sistemas com memória virtual* (*Jobs* eram constantemente “trocados” entre memória e disco (“swap”).
- **Tempo de resposta deve ser baixo.**
- Número de **tarefas processadas /por tempo** também deve ser baixo.
- Arquitetura mais complexa e de propósito geral.



Lotes x Tempo Compartilhado

	Batch Multiprogramming	Time Sharing
Principal objetivo	Maximizar o uso do processador (throughput)	Minimizar o tempo de resposta (response time)
Fonte de diretivas para o sistema operacional	Job control language Comandos providos com o job	Comandos via terminal

Sistemas de Tempo Real

- Sistemas dedicados a uma classe de aplicações críticas (ex. controle e medição)
 - Ex: sistemas de controle industriais, sistemas robóticos, sistemas de realidade virtual, automação doméstica, etc.
 - Ex: QNX, CMX-RTX, LynxOS, RTMX, etc.
- Possuem restrições temporais bem definidas (**o tempo de resposta** é considerado crítico).
- **Hard Real-Time**
 - Geralmente implementado por sistemas operacionais específicos para esse fim
- **Soft Real-Time**
 - Para aplicações que requerem características avançadas de sistemas operacionais (ex: multimídia, realidade virtual), mas que não são críticas

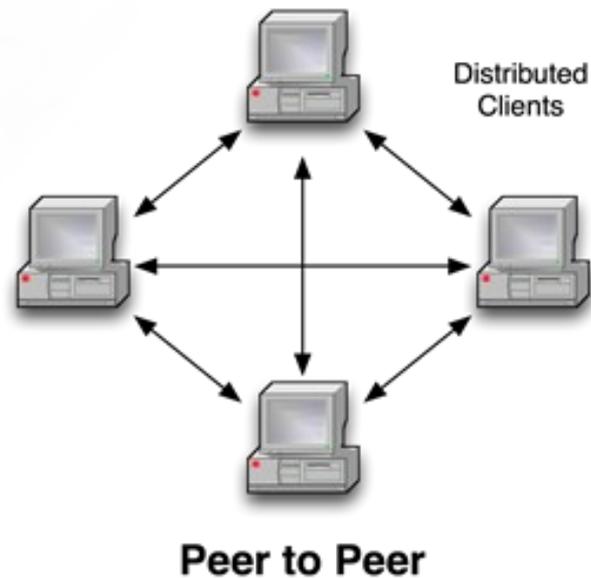
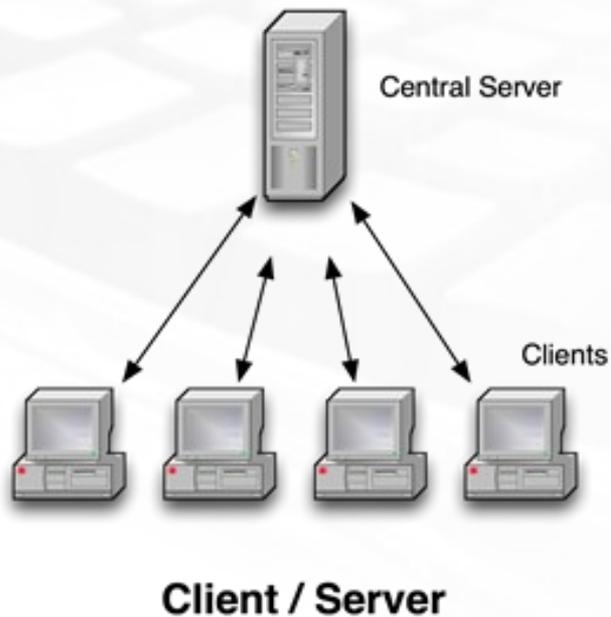
Evoluindo os Sistemas Operacionais

- **Sistemas Distribuídos**
- **Sistemas Paralelos**

Sistemas Distribuídos (1)

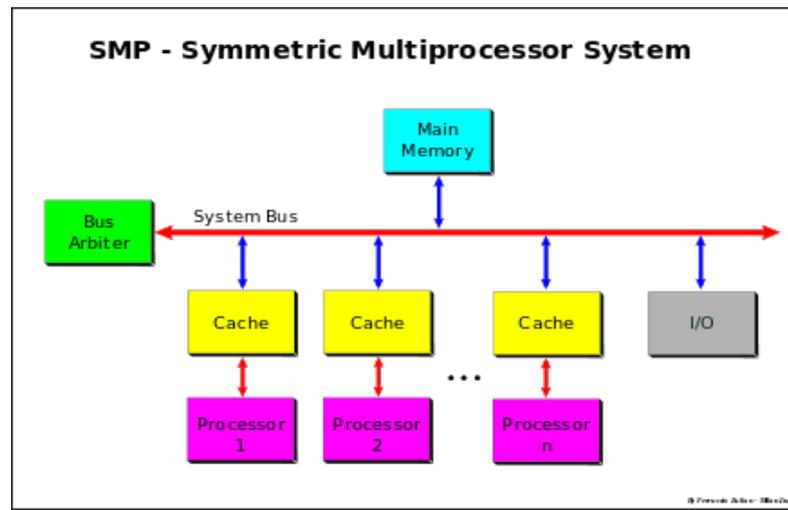
- Requer uma infra-estrutura física de redes (LAN, WAN) e suporte de protocolos de comunicação (TCP/IP).
- A realização de uma tarefa é distribuída entre vários nós da rede.
- Vantagens:
 - Compartilhamento de recursos;
 - Balanceamento de carga;
 - Aumento da velocidade de computação;
 - Maior confiabilidade.
- É um sistema **fracamente acoplado (Loosely coupled)**
 - Cada nó é uma máquina independente, com seu próprio S.O. e recursos de HW
- Diferentes arquiteturas
 - Cliente-Servidor
 - Peer-to-Peer

Sistemas Distribuídos (2)



Sistemas Paralelos

- As CPUs compartilham o barramento, memória e relógio, além de I/O (sistemas **fortemente acoplados** - ***tightly coupled systems***):
- Principais vantagens:
 - Aumento da vazão (*throughput*).
 - Economia de escala;
 - Aumento da confiabilidade.
- Symmetric multiprocessing (SMP)
 - Cada processador roda uma cópia idêntica do sistema operacional.
 - Não existe relação de mestre-escravo.
 - A maioria dos sistemas operacionais modernos suporta SMP



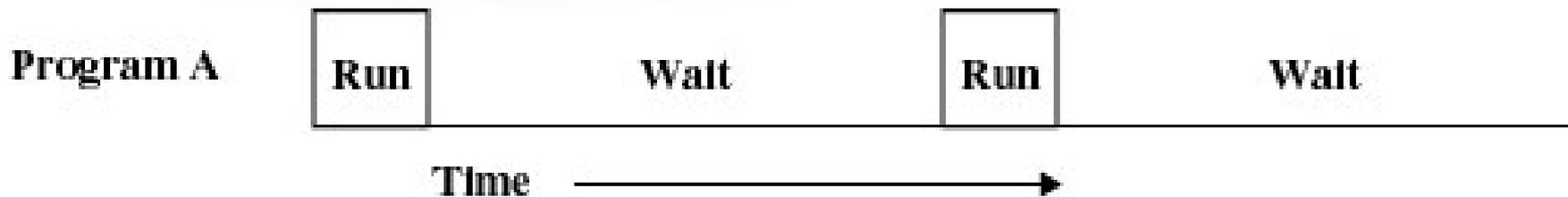
Classificação

- Quanto ao número de usuários:
 - **Monousuário**
 - **Multiusuário**

- Quanto ao número de tarefas:
 - **Monotarefa:**
 - Não suportam multiprogramação
 - **Multitarefa:**
 - **MULTIPROGRAMAÇÃO + Time Sharing !!!**

Mono x Multiprogramação (com *time sharing*)

- Suponha um programa que processa um arquivo de registros e que executa, em média, 100 instruções de máquina por registro.
 - Ler um registro 0,0015 seg
 - Executar 100 instruções 0,0001 seg
 - Gravar um registro 0,0015 seg
- Percentagem de utilização da UCP:
 - $U = 0,0001 / 0,0031 = 0,032 = 3,2\%$



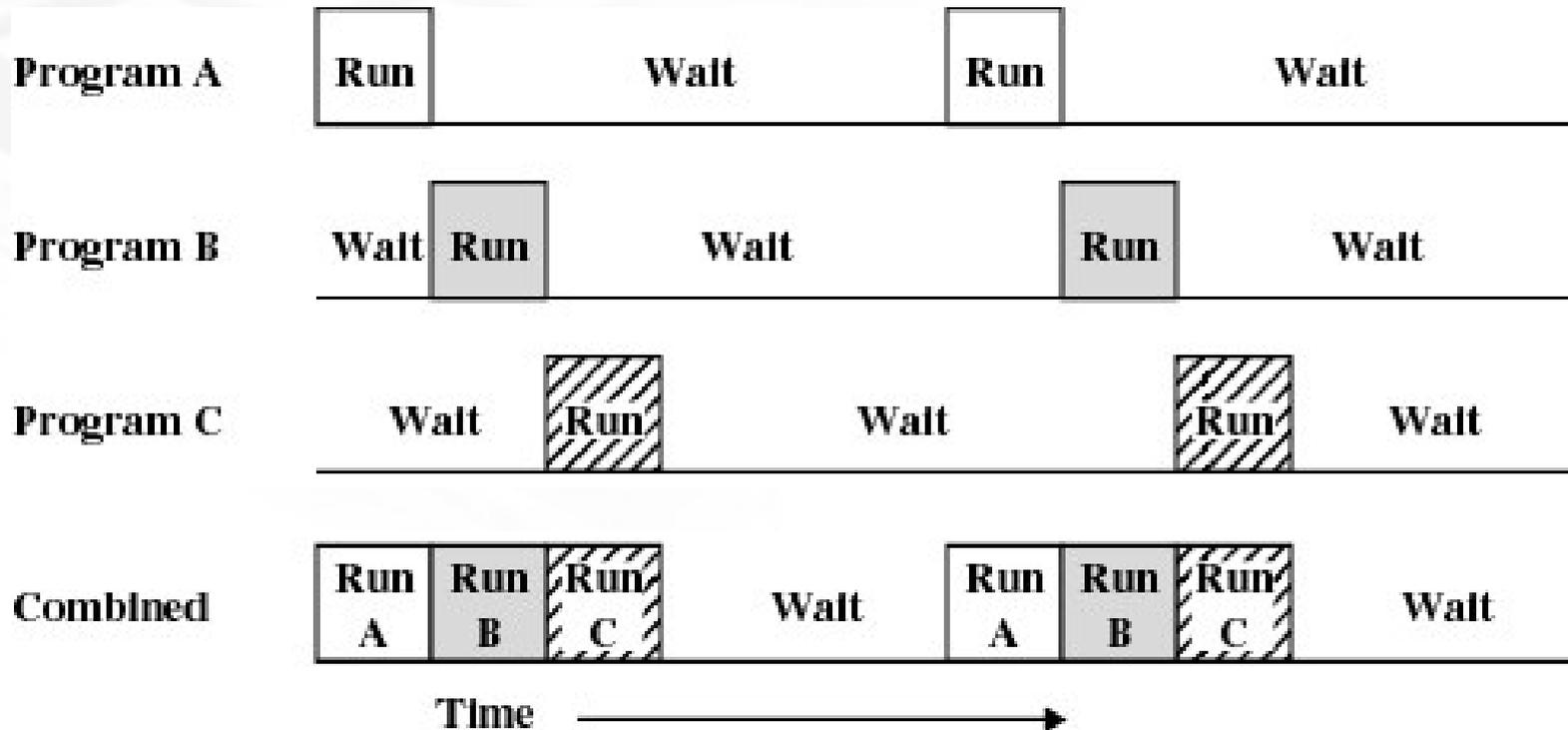
(a) Uniprogramming

Monoprogramação

- Os recursos computacionais estão inteiramente dedicados a um único programa/tarefa.
- A UCP fica ociosa durante muito tempo enquanto o programa aguarda por um evento (digitação de um dado, leitura do disco, etc.).
- A memória principal é subutilizada caso o programa não a preencha totalmente.
- Os periféricos são dedicados a um único usuário.
- Não existe grandes preocupações com a proteção de memória.
- A complexidade de implementação é relativamente baixa.

Multiprogramação (com *time sharing*)

- Vários programas competem pelos recursos do sistema.



(c) Multiprogramming with three programs

Multiprogramação (cont.)

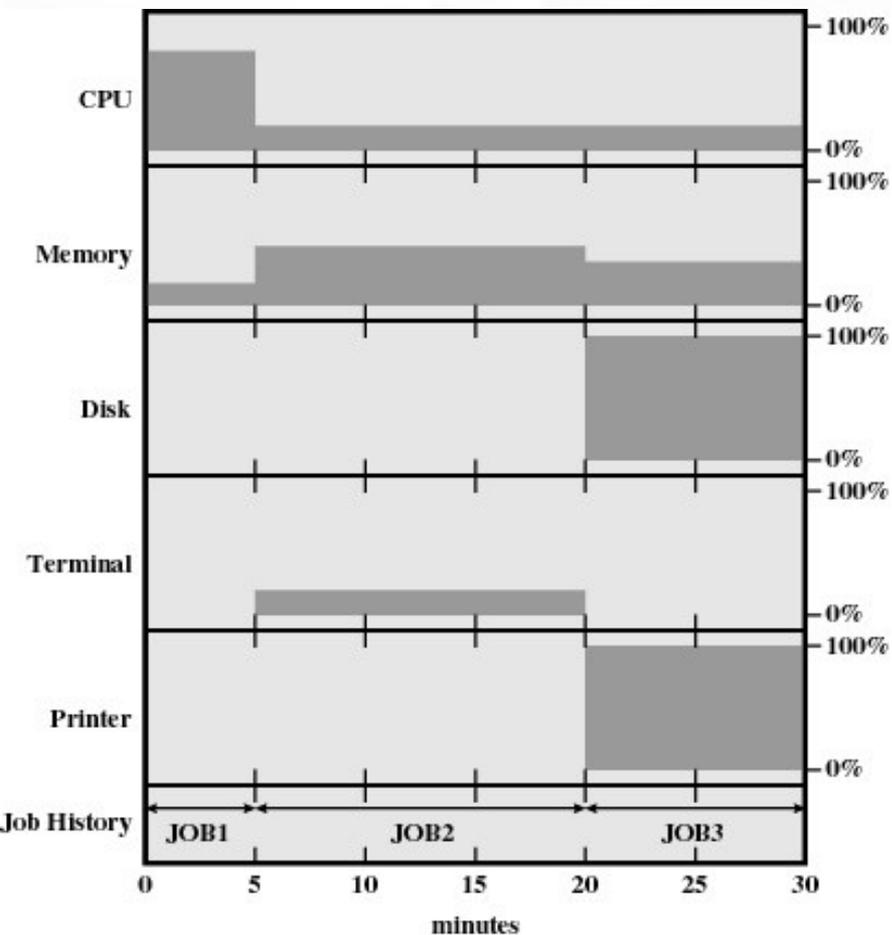
- O objetivo é manter mais de um programa em execução “simultaneamente”, dando a ilusão de que cada programa/usuário tem a máquina dedicada para si.
- A idéia é tirar proveito do tempo ocioso da UCP durante as operações de E/S. Enquanto um programa espera por uma operação de leitura ou escrita os outros podem estar sendo processados no mesmo intervalo.
 - Maximização do uso do processador e da memória.
 - Maior taxa de utilização do sistema como um todo (redução do custo total máquina/homem).
- Suporte de hardware:
 - Proteção de memória
 - Mecanismo de interrupção (sinalização de eventos).
 - Discos magnéticos (acesso randômico aos programas, melhor desempenho em operações de E/S) para implementação de memória virtual

Exemplo ⁽¹⁾

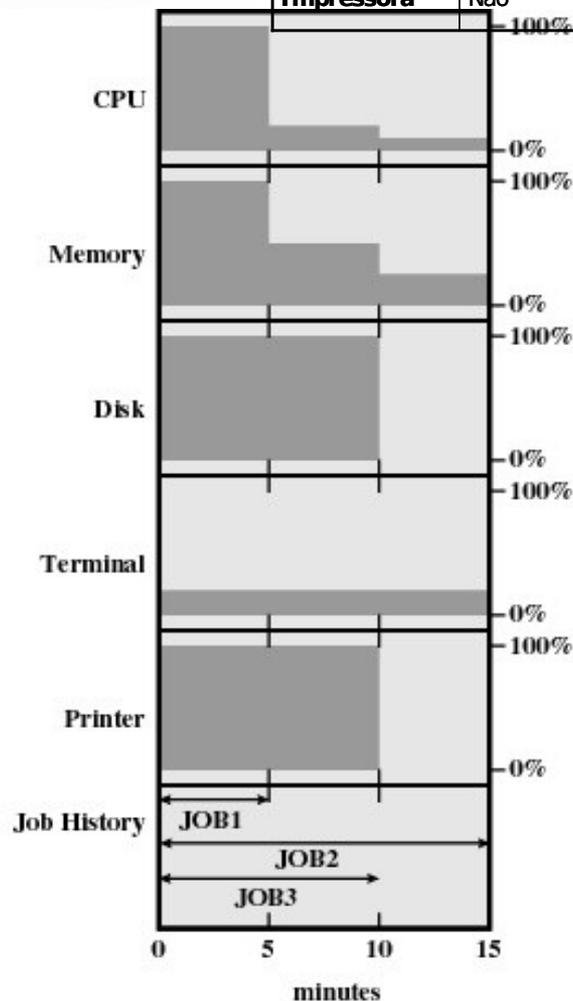
	JOB1	JOB2	JOB3
Tipo de Job	Muita UCP	Muita E/S	Muita E/S
Duração	5 min	15 min	10 min
Memória	50K	100K	80K
Disco	Não	Não	Sim
Terminal	Não	Sim	Não
Impressora	Não	Não	Sim

	JOB1	JOB2	JOB3
Tipo de Job	Muita UCP	Muita E/S	Muita E/S
Duração	5 min	15 min	10 min
Memória	50K	100K	80K
Disco	Não	Não	Sim
Terminal	Não	Sim	Não
Impressora	Não	Não	Sim

Exemplo (2)



(a) Uniprogramming



(b) Multiprogramming

Exemplo (3)

	Monoprogramação	Multiprogramação
Processor use	17%	33%
Memory use	33%	67%
Disk use	33%	67%
Printer use	33%	67%
Elapsed time	30 min.	15 min.
Throughput rate	6 jobs/hr	12 jobs/hr
Mean response time	18 min.	10 min.