

A large, faint, grayscale image of a computer keyboard is positioned in the background, angled from the bottom left towards the top right, creating a sense of depth and perspective.

# Sistemas Operacionais

Introdução

## Informações Gerais

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

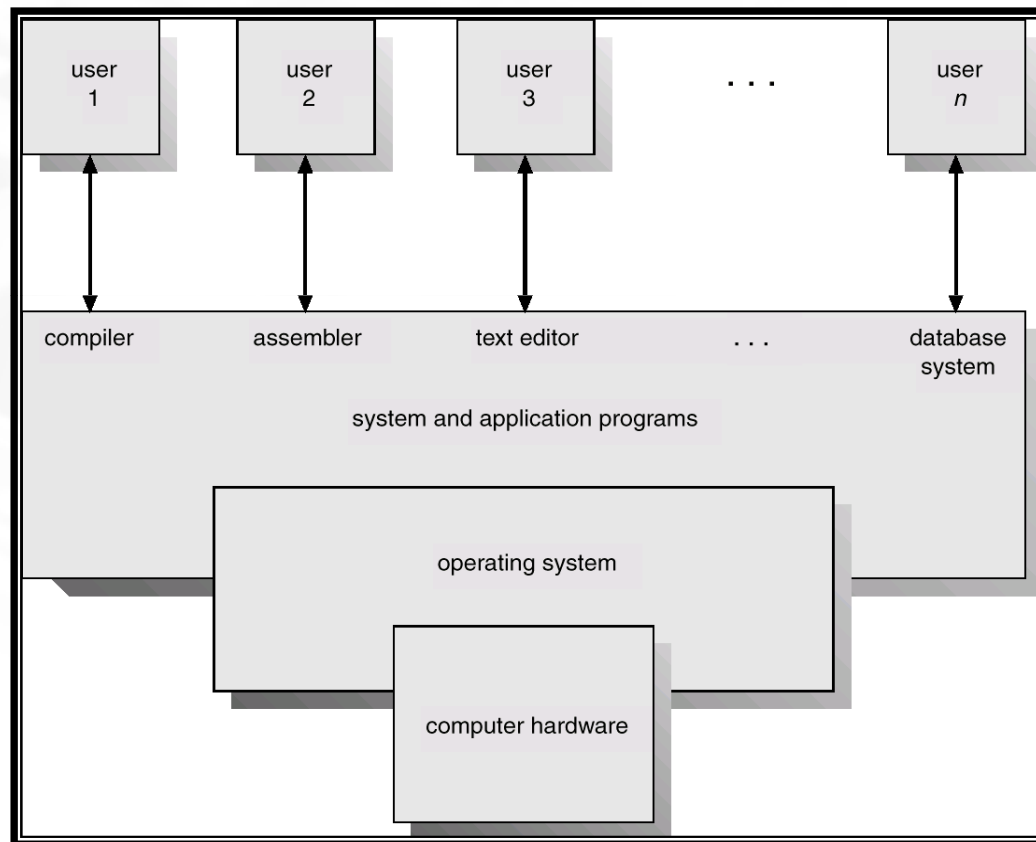
## Objetivo do Curso

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

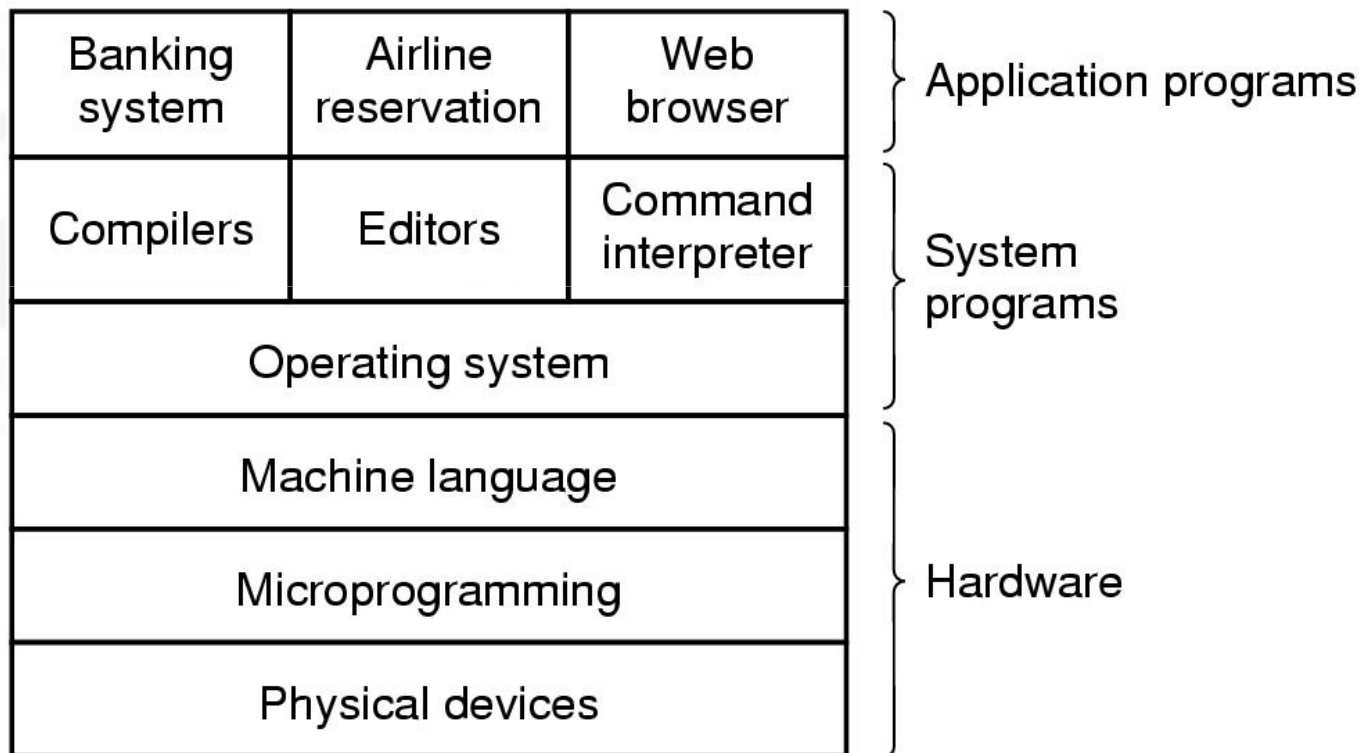
# Sistema de Computação

- 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)



## Visão Abstrata (2)



## Fato

- O hardware de um computador, sozinho, não fornece um ambiente simples, flexível e adequado para o desenvolvimento e uso dos programas de aplicação dos usuários.

## 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 (1)

- 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

## Definição (2)

- “A program that controls the execution of application programs.”
- “An interface between applications and hardware.”
- “Programa que age como um intermediário entre o usuário de um computador e o hardware deste computador”.

## Definição (3)

- “Resource allocator”
  - Manages and allocates resources.
- “Control program”
  - Controls the execution of user programs and operations of I/O devices .
- “Kernel”
  - The one program running at all times (all else being application programs).

## 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 e prover uma série de serviços (ex: sistema de arquivos, sistema de E/S) e uma interface de programação para acesso aos mesmos”.

## 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 E/S
  - Responsável pelo controle e execução de operações de E/S e otimização do uso dos periféricos.
- Sistema de Arquivos
  - Responsável pelo acesso e integridade dos dados residentes na memória secundária.
- Processador de Comandos (shell) / Interface com o Usuário
  - Responsável pela interface conversacional com o usuário.

## Histórico (1)

- No início ... (inexistência de S.O.)
  - O usuário é programador e operador da máquina ao mesmo tempo. Muita intervenção humana no processamento das tarefas.
  - *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.
- A esse objetivo se somou, em seguida, um segundo, de fundamental importância: a a otimização do uso dos recursos de hardware pelos programas.

## Histórico (2)

- Tal como nas arquiteturas de hardware, os sistemas operacionais também passam por um processo evolutivo, classificado em gerações.
- O surgimento dos S.O. data da segunda geração de computadores, na década de 50.

## Histórico (3)

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. 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. 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/Vista/7, Palm OS, Solaris, Unix SVR4, Android, etc.	HW para multiprocessamento simétrico, HW para computação móvel e ubíqua.



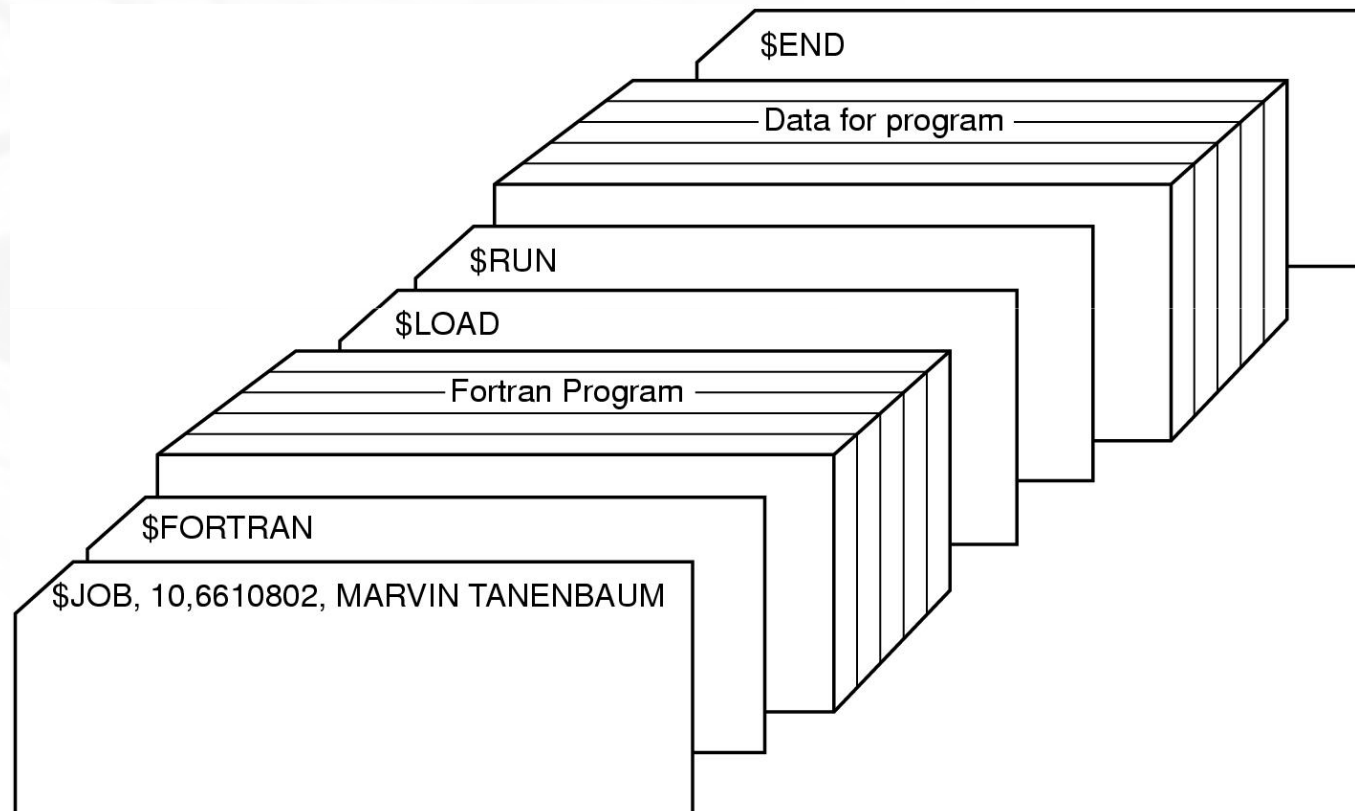
## Tipos de Sistemas Operacionais

- Sistemas de Lotes (Batch)
- Sistemas de Tempo Compartilhado (Time Sharing)
- Sistemas de Tempo Real (Real-Time)
- Sistemas *Mainframes*
- Sistemas *Desktop*
- Sistemas Distribuídos
- Sistemas Paralelos
- Sistemas *Handheld (mobile devices)*
- Sistemas embarcados/computação ubíqua/redes *ad hoc*

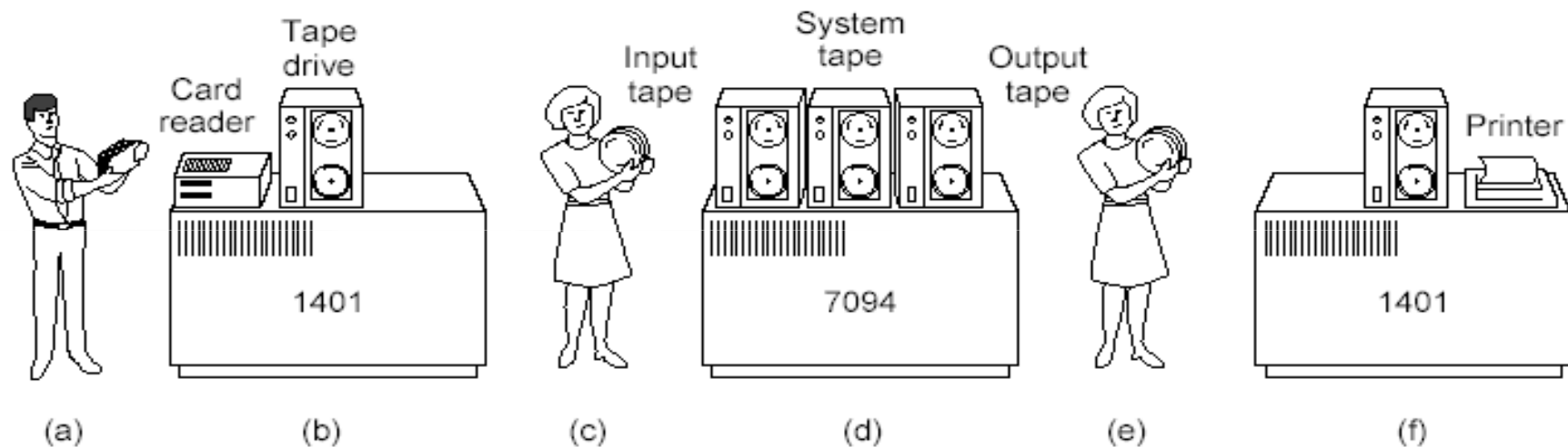
## Sistemas de Lotes (1)

- Primeiro tipo de sistema operacional (usuário e operador passam a ser entidades diferentes).
- *Job* é o conceito básico num sistema de lotes, sendo 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 sequencialidade, sem interrupção.
- Tempo médio de resposta pode ser muito alto (ex: presença de *jobs* grandes à frente de *jobs* pequenos).

## Sistemas de Lotes (2)

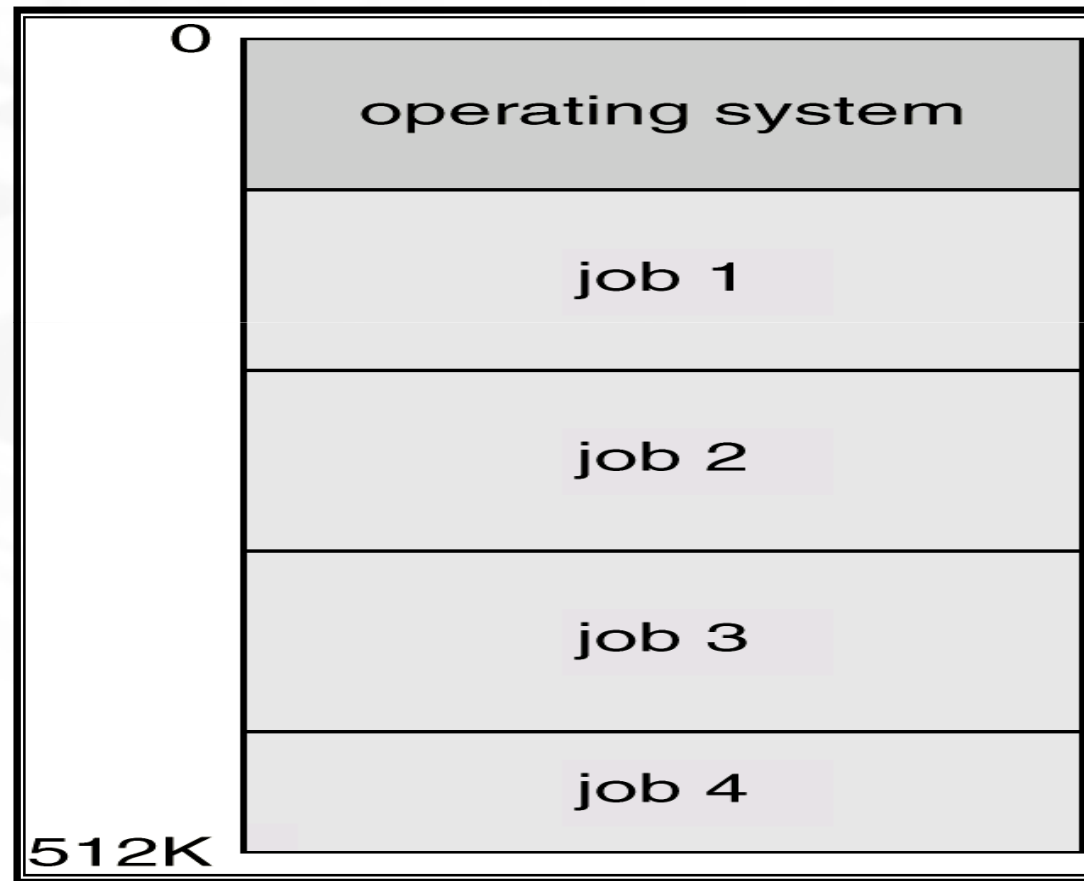


## Sistemas de Lotes (3)



**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



# Sistemas de Tempo Compartilhado

- A capacidade e o tempo de processamento da máquina é dividida entre múltiplos usuários, que acessam o sistema através de terminais.
- Usuário dirige a sua tarefa. Seus comandos são interpretados e executados em seguida (processamento *on-line*).
- Uso da técnica de multiprogramação para a manipulação de múltiplos *jobs* interativos.
- *Jobs* são constantemente “trocados” entre memória e disco (“swap”).
- Tempo de resposta é baixo.
- Número de tarefas processadas/tempo é baixo.
- Arquitetura mais complexa e de propósitos gerais.

## Lotes x Tempo Compartilhado

	<b>Batch Multiprogramming</b>	<b>Time Sharing</b>
Principal objetivo	Maximizar o uso do processador	Minimizar o tempo de resposta
Fonte de diretivas para o sistema operacional	Job control language Comandos providos com o job	Comandos via terminal

## Sistemas de Tempo Real (1)

- Sistemas dedicados a uma classe de aplicações, tipicamente de controle e medição, que exigem monitoramento contínuo de instrumentos.
  - 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).
- Pode ser *hard* ou *soft* real-time.



## Sistemas de Tempo Real (2)

### ■ *Hard Real-Time*

- Memória secundária limitada ou ausente (dados armazenados em memória ROM)
- Característica normalmente não suportada em sistemas operacionais de propósitos gerais.

### ■ *Soft Real-Time*

- Utilização limitada em controle industrial e robótica.
- Útil para aplicações que requerem características avançadas de sistemas operacionais (ex: multimídia, realidade virtual).

## 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 adotar tecnologia desenvolvida para S.O. de maior porte.
- Pode rodar diferentes tipos de S.O. (Windows, MacOS, UNIX, Linux, etc.)

## Sistemas Distribuídos (1)

- Conjunto de computadores interconectados de forma a possibilitar a execução de um serviço.
- 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.
- A existência de várias máquinas é transparente, isto é, o software fornece uma visão única do sistema.
- Vantagens:
  - Compartilhamento de recursos;
  - Balanceamento de carga;
  - Aumento da velocidade de computação;
  - Maior confiabilidade.

## Sistemas Distribuídos (2)

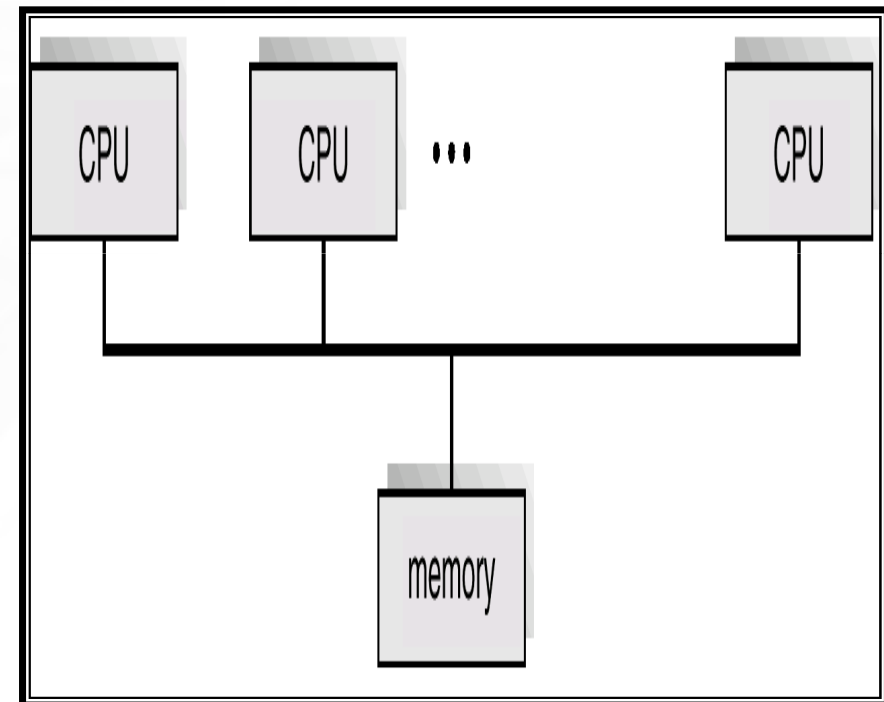
- Sistema fracamente acoplado (*Loosely coupled system*):
  - Cada nó é uma máquina independente, com seu próprio S.O. e sua própria memória local, comunicando-se com os outros através das várias linhas de comunicação.
- Arquitetura Cliente-Servidor:
  - Mudança do paradigma de arquitetura centralizada, provocada pelo barateamento e aumento da capacidade de processamento e armazenamento dos PCs.
  - Servidores são configurados para satisfazer as consultas dos sistemas clientes (servidor de arquivos, servidor de mail, servidor de ftp, etc.).

## Sistemas Paralelos(1)

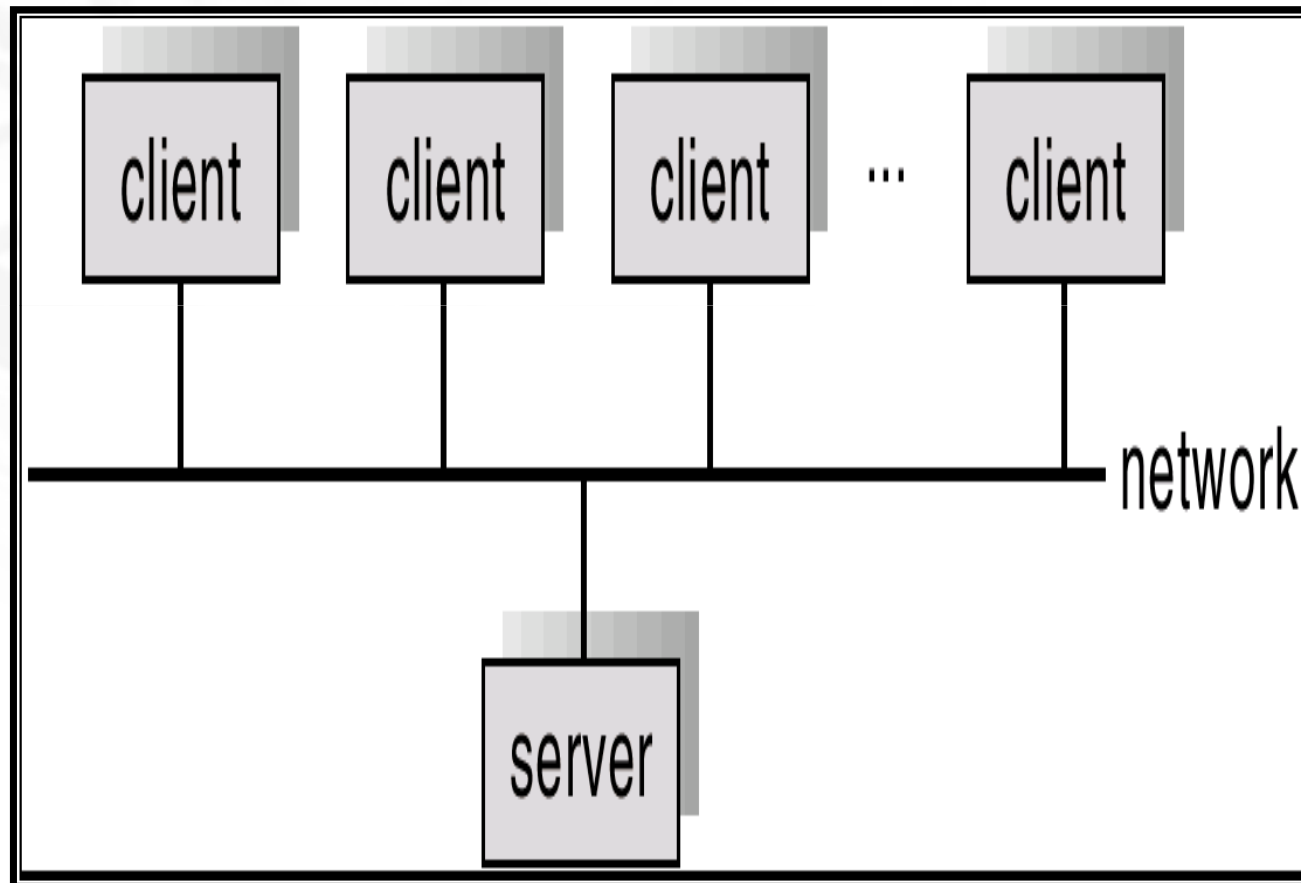
- São sistemas com mais de uma UCP atuando de modo colaborativo (“*multiprocessor systems*”).
- Os processadores compartilham o barramento, memória e relógio, além de dispositivos periféricos (são sistemas fortemente acoplados – “*tightly coupled systems*”):
- Principais vantagens:
  - Aumento da vazão (“*throughput*”).
  - Economia de escala;
  - Aumento da confiabilidade.

## Sistemas Paralelos(2)

- *Symmetric multiprocessing (SMP)*
  - Cada processador roda uma cópia idêntica do sistema operacional.
  - Vários processos podem ser executados em paralelo, sem perda de desempenho para o sistema.
  - Todos os processadores são pares, não existindo relação de mestre-escravo.
  - A maioria dos sistemas operacionais modernos suporta SMP, através do conceito de *multithreading* (Windows, Solaris, OS/2, Linux).



## Sistemas Distribuídos (3)





## Sistemas *Handheld*

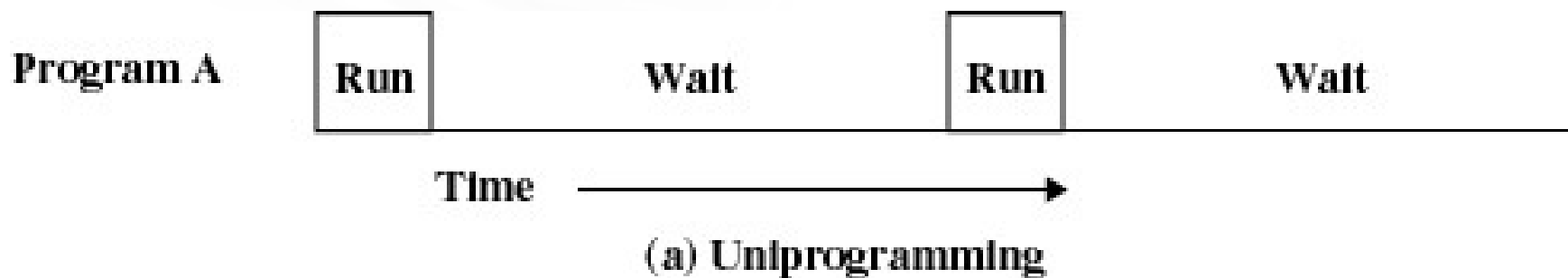
- Sistemas operacionais adaptados para dispositivos móveis, como PDAs – *Personal Digital Assistants*, Tablets e Smartphones.
- Memória limitada, processador mais lento e display de pequenas dimensões.
- S.O. e aplicações projetados para minimizar o uso do processador (redução do consumo da bateria).
- Uso possível de tecnologias *wireless*, como Bluetooth e Wi-fi, para acesso remoto a e-mail e navegação Web.
- Câmeras, MP3 players e GPS são exemplos de acessórios que expandem a sua funcionalidade.

## Classificação

- Quanto ao número de usuários:
  - Monousuário:
    - Projetados para suportar um único usuário
    - Ex: MS-DOS, Windows 3x, Windows 9x.
  - Multiusuário:
    - Projetados para suportar várias sessões de usuários
    - Ex: Windows NT(2000), UNIX.
- Quanto ao número de tarefas:
  - Monotarefa:
    - Capazes de executar apenas uma tarefa (um aplicativo) de cada vez.
    - Ex: MS-DOS
  - Multitarefa:
    - Capazes de executar várias atividades simultaneamente, como uma compilação e um processamento de texto
    - Ex: Windows, OS/2, Unix.

## Mono x Multiprogramação

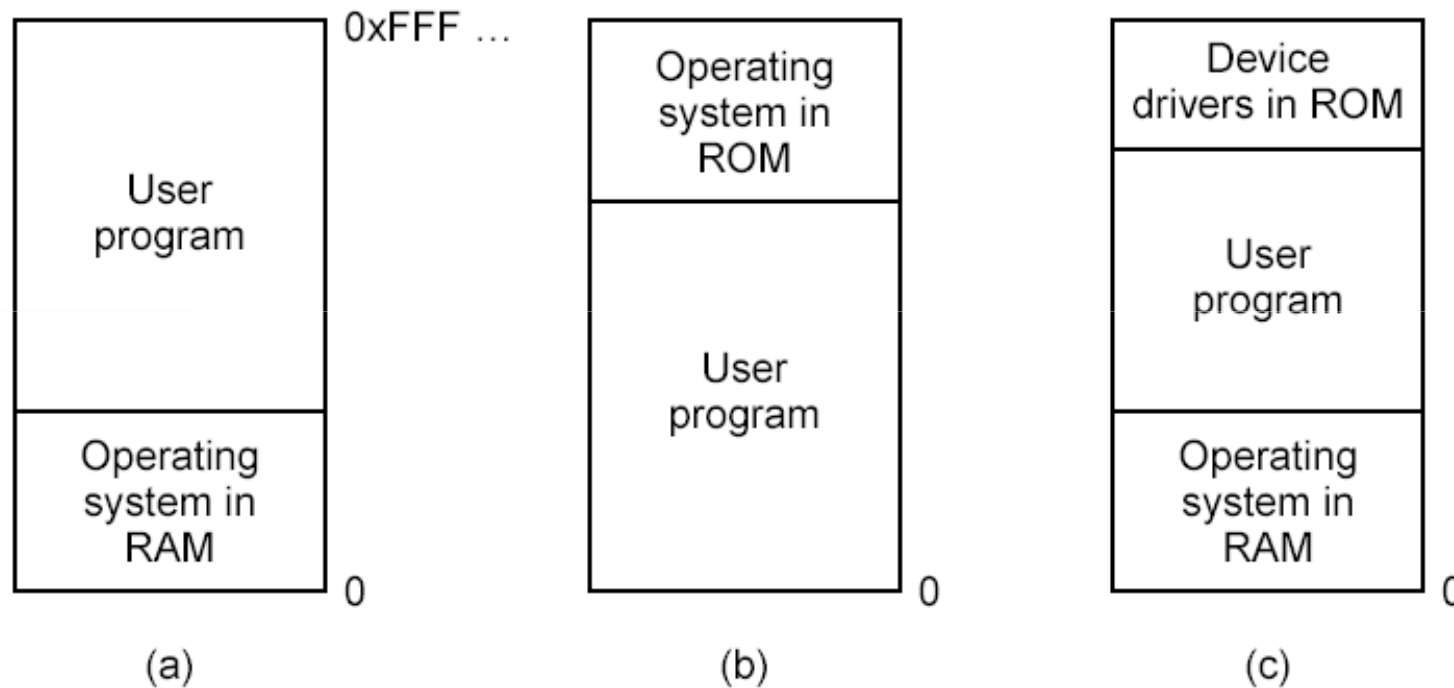
- Suponha um programa que processa um arquivo de registros e que executa, em média, 100 instruções de máquina por registro.
  - Ler um registro 0,0015 seg
  - Executar 100 instruções 0,0001 seg
  - Gravar um registro 0,0015 seg
- Percentagem de utilização da UCP:
  - $U = 0,0001 / 0,0031 = 0,032 = 3,2\%$



## 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.

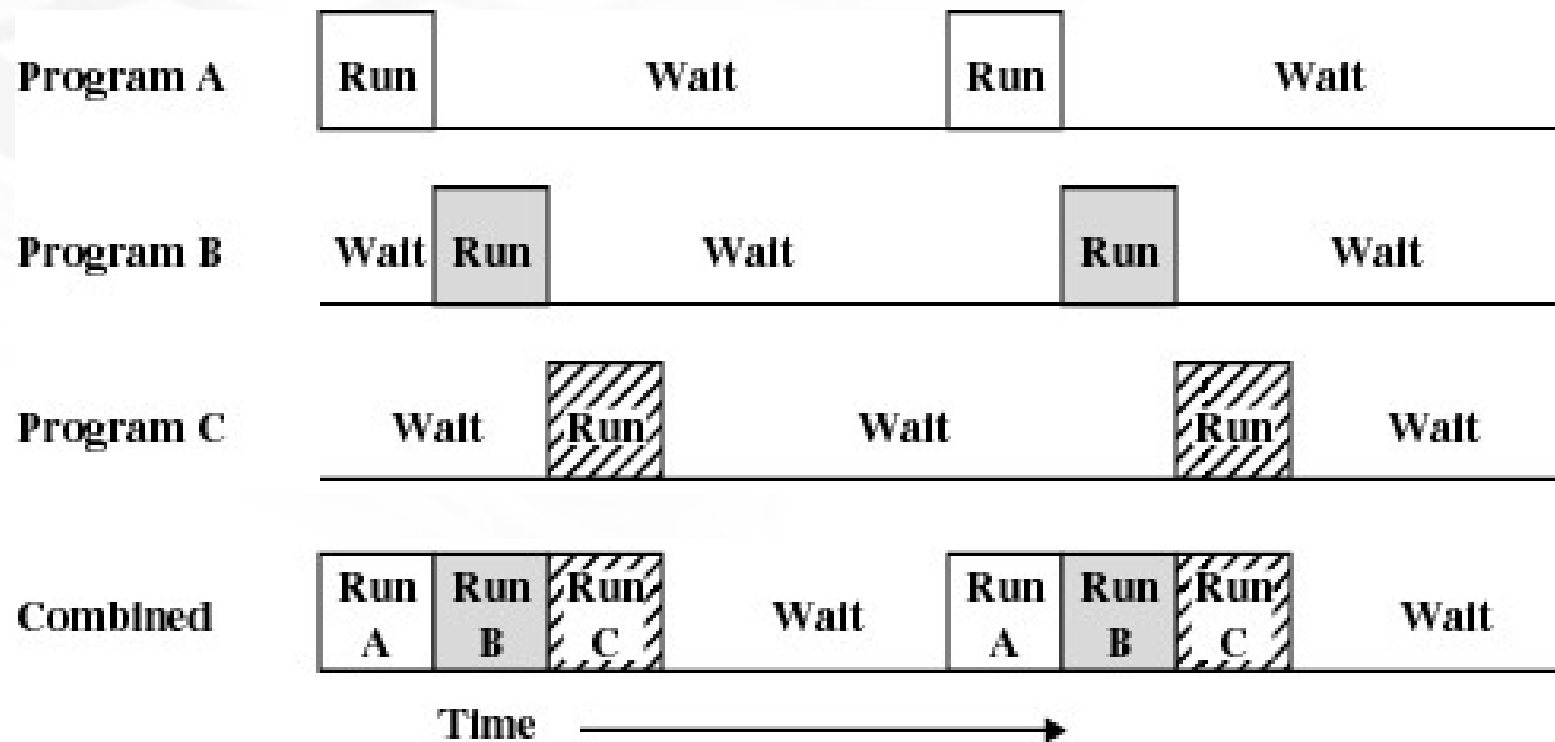
## Monoprogramação (cont.)



**Figure 4-1.** Three simple ways of organizing memory with an operating system and one user process. Other possibilities also exist.

# Multiprogramação

- Vários programas competem pelos recursos do sistema.

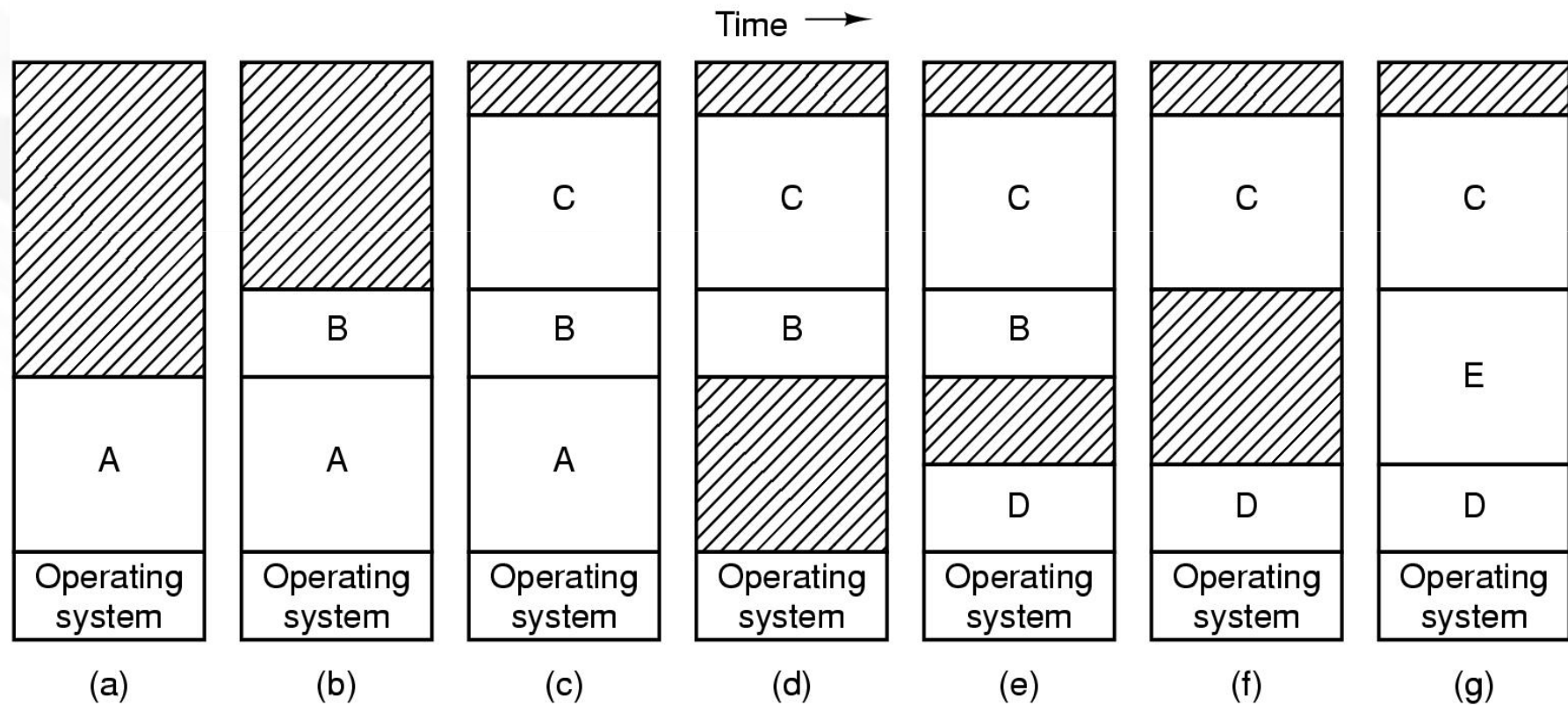


(c) Multiprogramming with three programs

## 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

# Multiprogramação (cont.)



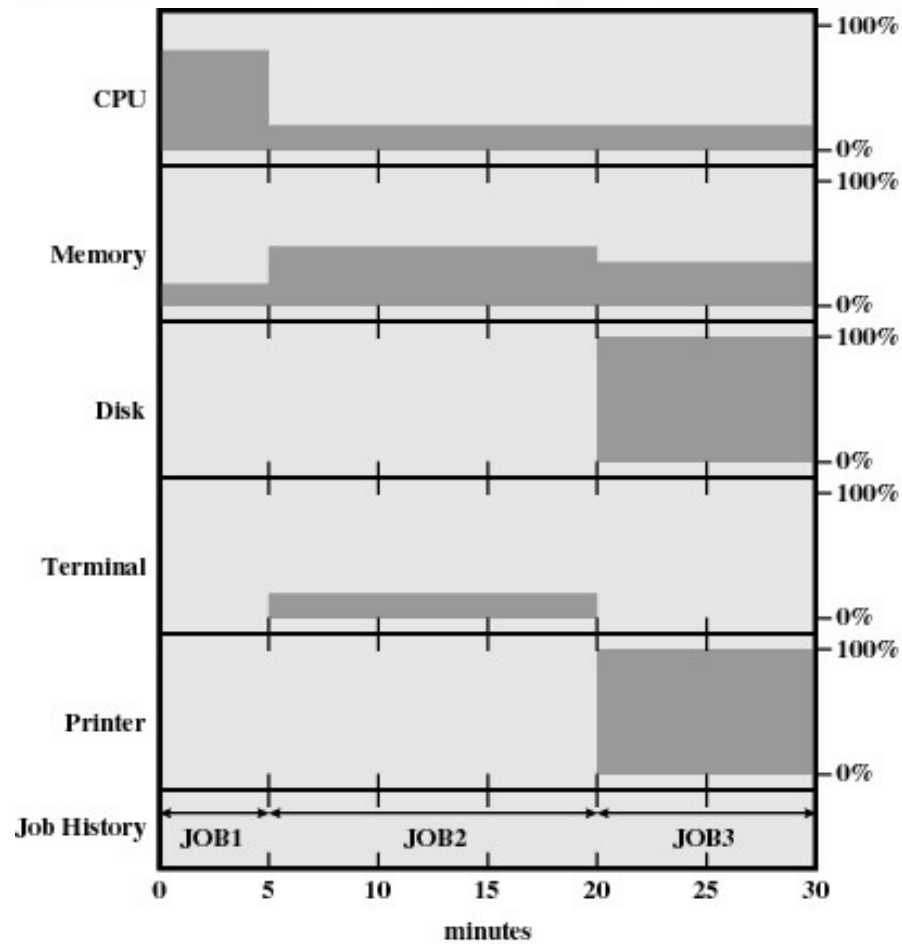


## Exemplo (1)

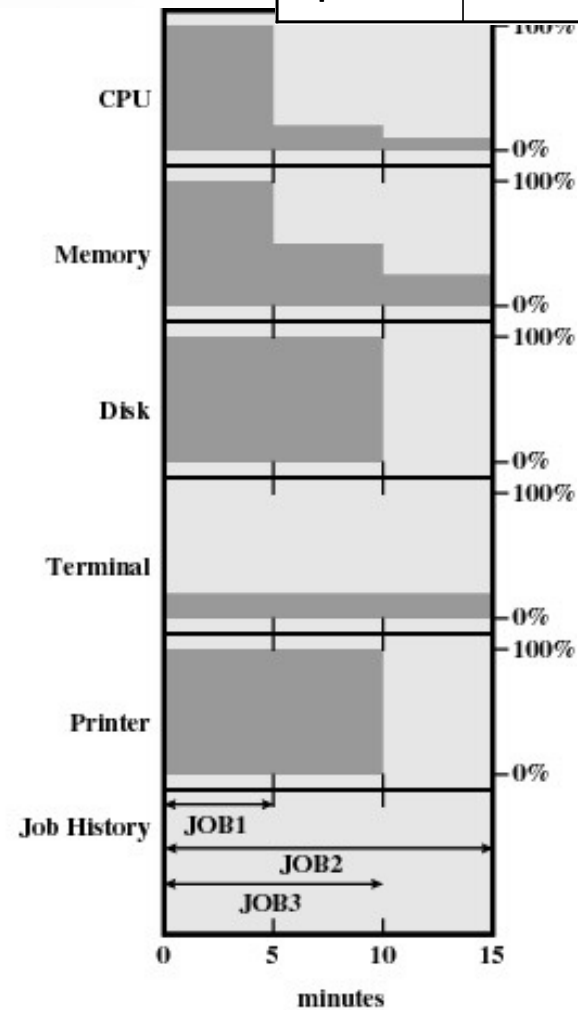
	JOB1	JOB2	JOB3
<b>Tipo de Job</b>	Muita UCP	Muita E/S	Muita E/S
<b>Duração</b>	5 min	15 min	10 min
<b>Memória</b>	50K	100K	80K
<b>Disco</b>	Não	Não	Sim
<b>Terminal</b>	Não	Sim	Não
<b>Impressora</b>	Não	Não	Sim

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

## Tarefas

- Ler o capítulo 1 do livro texto (Deitel).
- Faça a lista de exercícios deste capítulo.
- Faça um dos trabalhos de pesquisa sugeridos ao final do capítulo.