

Endereçamento IP

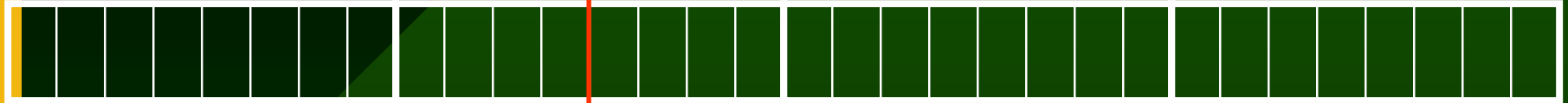
Histórico

- Iniciativa do Departamento de Defesa dos EUA
 - Gerenciamento Distribuído, livre de falhas pontuais
 - Ataque Nuclear ?
 - Atendia primeiro ao departamento de pesquisa e universidades
 - Depois os fornecedores;
 - Depois os terceiros;
 - Depois o MUNDO...

A . B . C . D



- 32 bits = 2^{32} endereços possíveis!
 - Cada conjunto de oito bits na época era chamado de “octeto”, e não de Byte;
 - Essa denominação é utilizada até hoje.
- Dividido em duas partes: REDE e HOST
- Máscara Identificava onde estava a divisão



Rede (*network*)

Dispositivo (*host*)

Algumas regrinhas importantes:

- Não existem dois endereços de rede válidos iguais;
- Dentro de uma determinada rede, não existem dois endereços de host iguais;
- Todo endereço IP em que todos os bits de host são iguais a 0 designa um “endereço de rede”;
- Todo endereço IP em que todos os bits de host são iguais a 1 designa um “endereço de broadcast”.

Endereço IP - 32 bits (quatro octetos)

A	A	A	A	A	A	A	A	B	B	B	B	B	B	B	C	C	C	C	C	C	C	D	D	D	D	D	D	D			
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

Determinam o endereço de rede, *host* e sub-rede



Os primeiros bits (entre 1 e 5) determinam a classe.

A . B . C . D



Endereço
de
Rede

Endereço
de
host

- Endereços Classe A

- Identificados pela presença de um zero no primeiro bit dos quatro octetos;
- Atendiam a um número limitado de empresas (até 128 combinações);
- Cada empresa podia ter até 2^{24} computadores identificados.

A . B . C . D

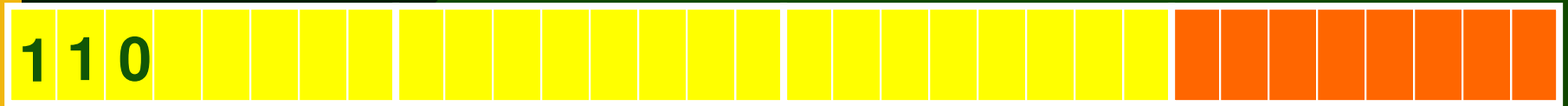


Endereço
de
Rede

Endereço
de
host

- Endereços Classe B
 - Identificados pela presença de um zero no segundo bit dos quatro octetos;
 - Atendiam a um número médio e empresas (até 2^{14} combinações);
 - Cada empresa podia ter até 2^{16} computadores identificados.

A . B . C . D



Endereço
de
Rede

Endereço
de
host

- Endereços Classe C

- Identificados pela presença de um zero no terceiro bit dos quatro octetos;
- Atendiam a um número enorme de empresas (até 2^{21} combinações);
- Cada empresa podia ter até 256 computadores identificados.

A . B . C . D



Identifica
Classe D

Identificação do grupo
de multicast
(28 bits)

A . B . C . D



Identifica
Classe E

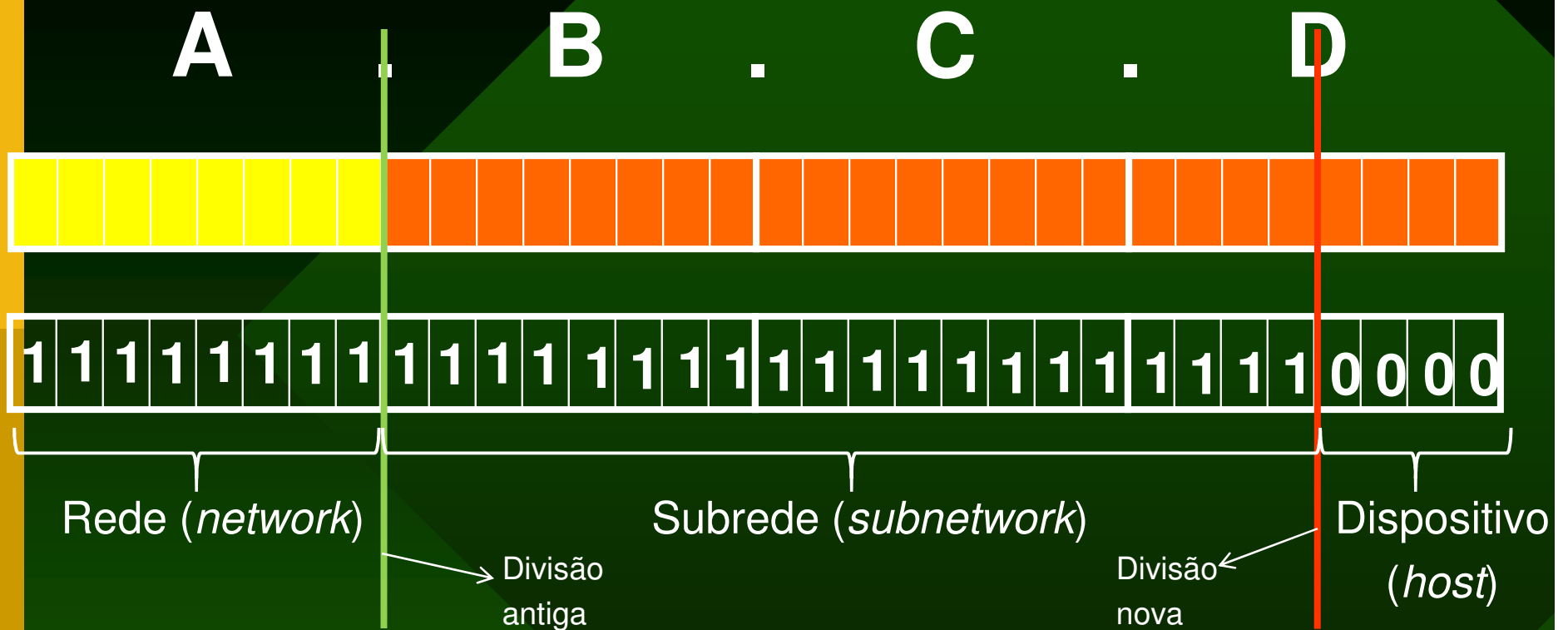
Classe E: testes e experimentos. Inválida
para endereçamento válidos na Internet.

Anos 90 : Surgimento dos Hackers

- Oferecer endereços válidos para usuários comuns era muito perigoso
 - Surgimento dos *proxies* (procuradores);
 - Necessidade muito menor de endereços válidos.
- Sub-redes
 - Máscaras voltam a ser úteis
 - Lembrem-se que a máscara não pode ser menor que a máscara padrão da classe !

Além das subredes ...

- É importante lembrar que o conceito de reservar sem uso as sub-redes com todos os bits identificadores da mesma iguais a 1 ou iguais a 0 caiu por terra a muito tempo ...
 - A [RFC950](#), de agosto de 1985, recomendava a reserva (“...*This means the values of all zeros and all ones in the subnet field should not be assigned to actual (physical) subnets...*”)
 - A [RFC1878](#), de dezembro de 1995, retirou esta exigência (“...*note that all-zeros and all-ones subnets are included...*”)



- O ambiente não exige mais tantas estações. Se a organização precisa de 10 hosts, por exemplo, basta reservar os 4 últimos bits !
- Bits restantes identificam a “sub-rede”

Além das subredes ...

- O conceito de VLSM (*Variable Length Subnet Masking*) envolve a implementação de diversas etapas de subdivisão dos endereços de rede;
- Os grupos maiores (formados nas primeiras subdivisões) atendem às redes maiores; os grupos menores às redes menores;
- O importante é que as subredes tenham tamanhos próximos das demandas localizadas por *hosts*.

Além das subredes ...

- O conceito de CIDR (*Classless InterDomain Routing*) envolve o chamado “agregado de rotas”;
- Utilizando endereços maiores, que podem conter diversas subredes, podemos reduzir as tabelas de roteamento dos roteadores.

Vamos fazer alguns exemplos?

- Visitem o site www.catspace.com
- O site contém listas de exercícios e respostas relacionadas a sub-redes

Introdução aos Sistemas Operacionais

- Definição
- O que é ?
- Exemplos
- Máquinas de Níveis

Definição

- “... um conjunto de rotinas executadas pelo processador, de forma semelhante aos programas dos usuários.”
- “... O SO tem por objetivo funcionar como uma interface entre o usuário e o computador, tornando sua utilização mais simples, rápida e segura”.
 - Francis Machado e Luiz Paulo Maia

Definição

- “... é um programa que atua como intermediário entre o usuário e o hardware de um computador.”
- “... deve propiciar um ambiente no qual o usuário possa executar programas de forma conveniente e eficiente”.
 - Silberschatz, Galvin e Gagne

O que é um Sistema Operacional?

- **O que é um S.O.?**
 - Funções
 - Responsabilidades
 - Transparência
 - Simplificação
 - Gerência
 - Compartilhamento
 - Otimização
 - Encapsulamento
 - Esconder Detalhes

O que é um Sistema Operacional?

- Primeiros computadores
 - Programação complexa
 - Exigia grande conhecimento do hardware e de linguagem de máquina
 - Solução:
 - Sistemas Operacionais
 - Encapsulamento
 - » Interação se tornou mais fácil, confiável e eficiente.

Exemplos

- Windows (Windows 7)
 - Usuário e servidores
- MAC OS X (Snow Leopard)
- Linux (Ubuntu, RedHat, SuSE)
- Unix (FreeBSD, SCOUnix, HP-UX, SunOS)

Exemplos

- Quais as principais diferenças?
 - Quem conhece mais de um sistema operacional?
 - Apresentar diferenças.

Máquina de níveis

- Computador como máquina de níveis ou camadas:
 - Nível 0 – *Hardware*;
 - Nível 1 – Sistema Operacional.
- *Hardware*:
 - Dispositivos físicos, microprogramação, linguagem de máquina.

Máquina de níveis

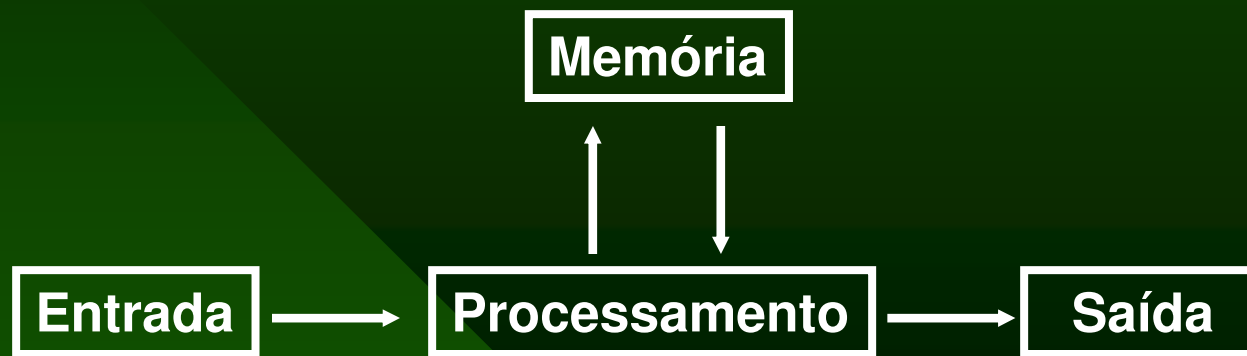
- *Software:*
 - Sistema operacional, utilitários e aplicativos;
 - Linguagem utilizada em cada um dos níveis varia da mais elementar (baixo nível) à mais sofisticada (alto nível).

Conceitos de HW e SW

Conceitos

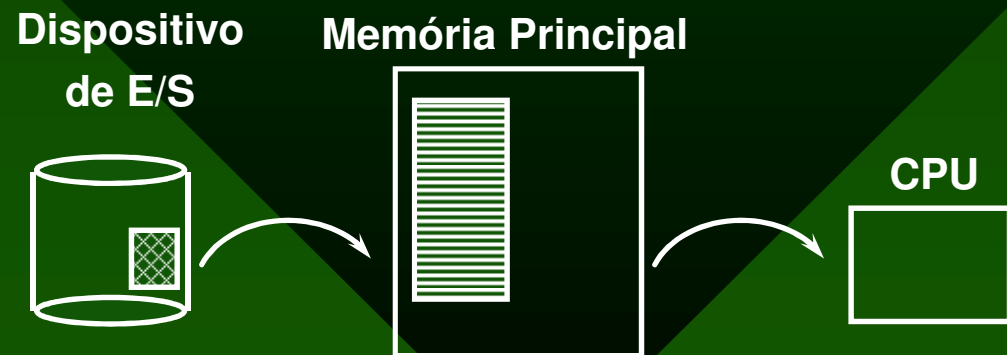
- *Hardware*
 - Três subsistemas básicos:
 - Unidade Central de Processamento;
 - Memória principal;
 - Dispositivos de entrada e saída.
 - Subsistemas são também chamados de unidades funcionais;
 - Implementações podem variar a depender da arquitetura.

Modelo de Computador



Modelo de Computador

- Execução de programas
 - Programa armazenado em dispositivo de E/S
 - Carga do programa na memória (SO)
 - Execução das instruções (uma a uma) na CPU



CPU

- Controla cada unidade funcional do sistema
- Execução das instruções dos programas
 - Aritméticas
 - Comparação
 - Movimentação

CPU

- **Conjunto de Instruções**
 - Instruções de máquinas disponíveis na CPU
 - Tipos e quantidade variam de acordo com a CPU
 - Compatibilidade de programas
 - RISC x CISC

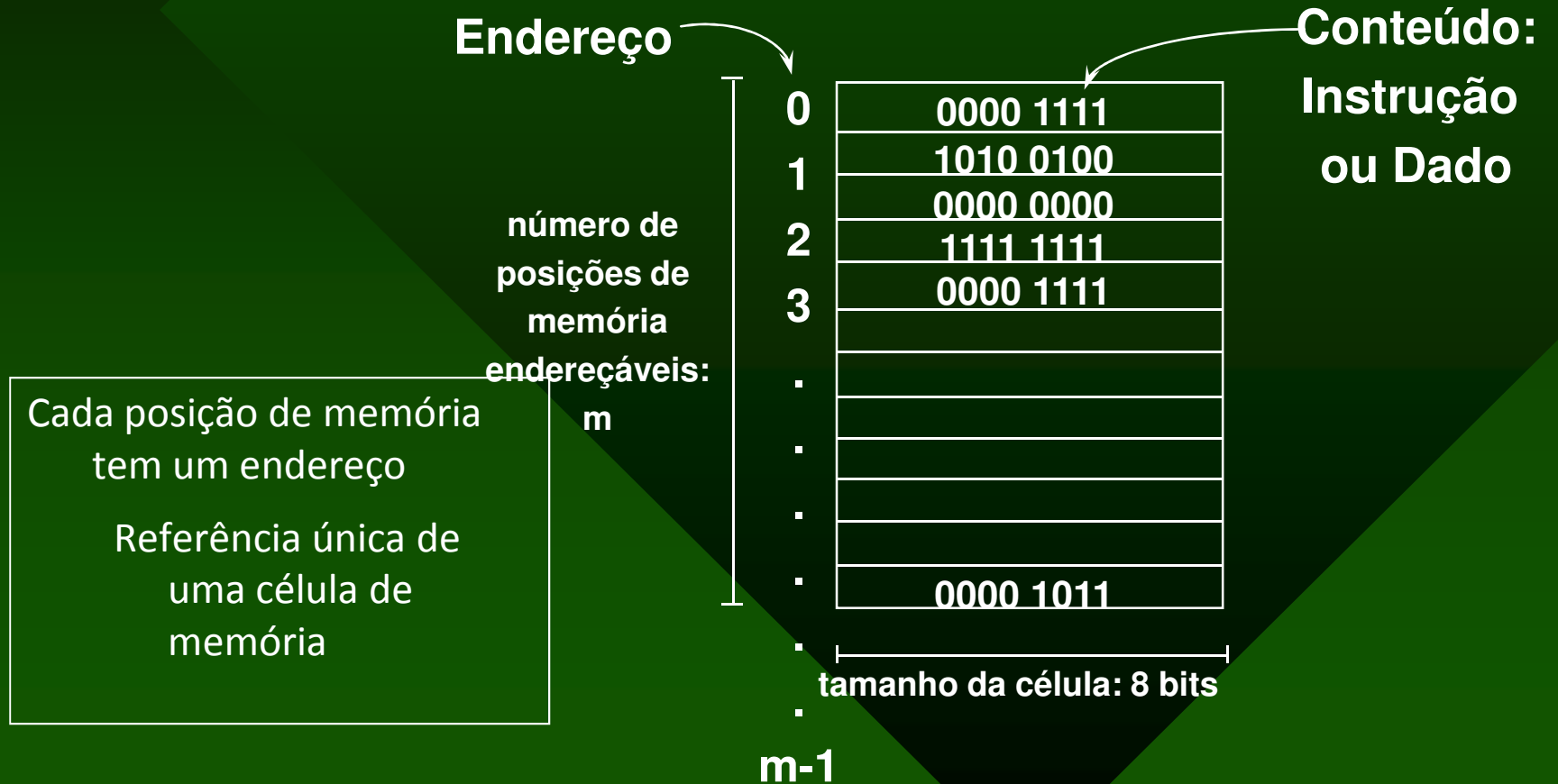
CPU

- **Desempenho de processadores**
 - Depende de:
 - » Conjunto de instruções disponível
 - » *clock*
 - » Tipo da aplicação: comercial x científica
 - *Benchmark*
 - » Usado para comparar processadores diferentes

Memória

- Memória Principal
 - Armazena informações (*bits*)
 - Programas (instruções)
 - Dados
 - Composta por várias unidades de acesso
 - Células
 - Posições de memória

Memória



Memória

- Interligação Memória - Processador
 - Registradores de uso específico:
 - Memory Address Register - MAR - REM
 - Memory Data Register - MDR - RDM

Memória

- Operação de Leitura da Memória
 - CPU armazena endereço da célula a ser lida no MAR
 - CPU gera sinal de controle indicando que a operação é de leitura da memória
 - Memória recupera informação armazenada na posição endereçada e coloca no barramento de dados, chegando ao MDR

Memória

Operação de Gravação na Memória

- CPU armazena endereço da célula a ser gravada no MAR
- CPU armazena a informação a ser gravada no MDR
- CPU gera sinal de controle indicando que a operação é de gravação na memória
- Memória armazena informação do barramento de dados na posição endereçada

Memória

- Memória Cache
 - Memória de alta velocidade
 - Localizada entre CPU e Memória Principal
 - Aumento de desempenho com custo razoável
 - Algoritmo
 - *Hit rate* - taxa de acertos
 - Interna x Externa

Memória

Maiores
Capacidade



Registradores

Memória Cache

Memória Principal

Memória Secundária

Menores
Custo e Velocidade de Acesso



Dispositivos de E/S

- Dispositivo de E/S
 - Comunicação com meio externo
 - Memória Secundária e Interface Homem-Máquina
- Barramentos
 - Conjunto de fios paralelos
 - Interliga CPU, Periféricos e Memória Principal

Barramentos

- Barramento de Dados
 - Número de *bits* por operação de E/S ou acesso à Memória
 - Processadores de 8, 16, 32 ou 64 bits
 - Fluxo Bi-direcional (leitura / gravação)
- Barramento de Endereços
 - Capacidade de Armazenamento = 2^n
 - Fluxo Uni-direcional (CPU → MEM ou E/S)

Barramentos

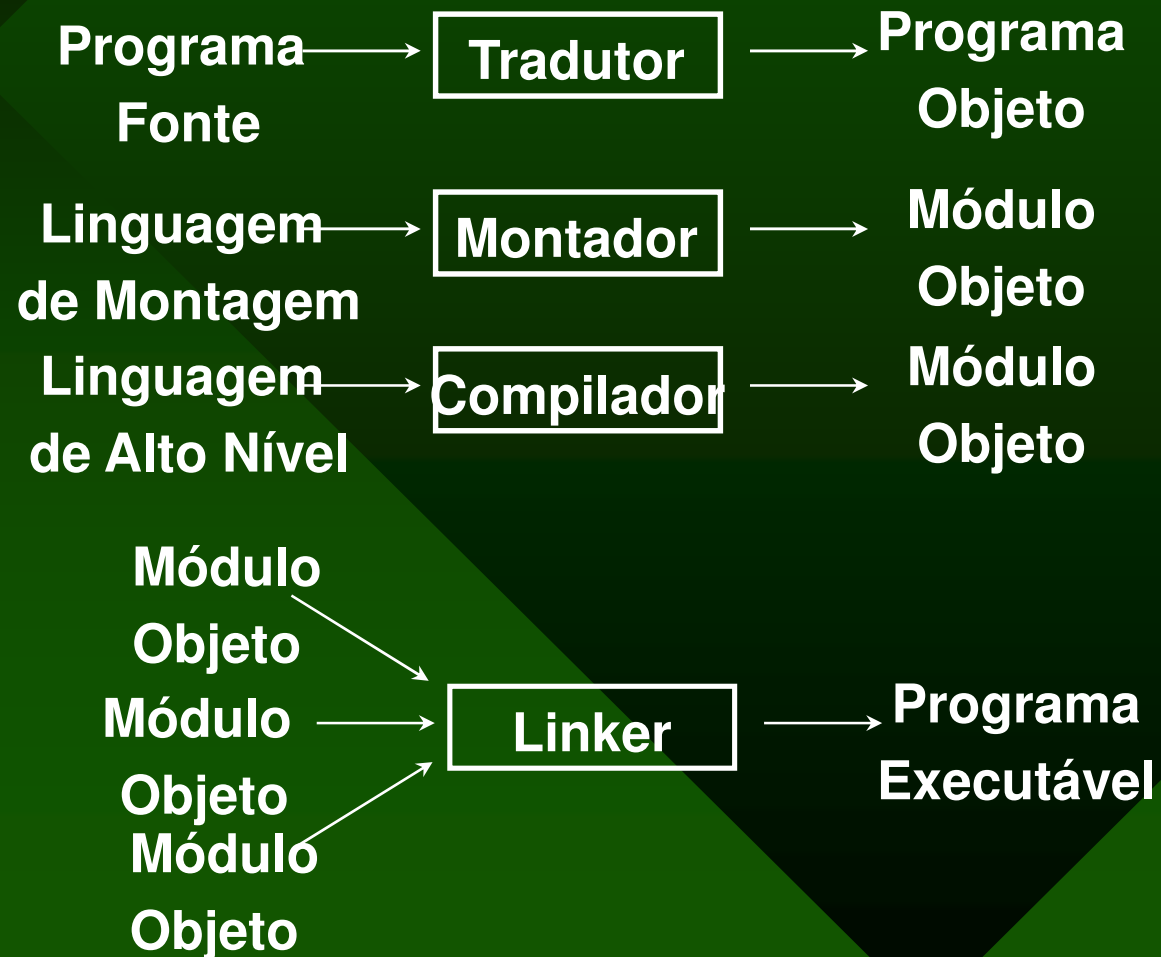
- Barramento de Controle
 - Diversos sinais de controle
 - *Read / Write*
 - Controle de Interrupções
 - Fluxo Uni ou Bi-direcional (depende do sinal de controle)

Conceitos de Software

Conceitos de Software

- Tradução
 - Necessidade de codificar instruções para máquina
 - Evolução
 - Programação via painéis
 - Linguagens de programação
 - Linguagens de Programação
 - Maior independência do HW
 - Maior facilidade de desenvolvimento / manutenção

Tradução



Tradução

- Linguagens de Montagem (*Assembly*)
 - Mnemônicos associados às instruções do processador
 - Específica para cada processador
 - Montadores (*Assembler*)

Tradução

- Linguagem de Alto Nível
 - Sem relação direta com o processador
 - Transportabilidade
 - Pascal, Fortran, Cobol, VB, Java
 - Compiladores
 - Interpretadores
 - Caso especial do Java: JVM

Tradução

- Módulos Objeto
 - Código de máquina não executável
 - Referências a módulos externos não resolvidas
- Módulos Executáveis
 - Códigos de máquina executáveis
 - Linkers
 - Bibliotecas
 - Relocação

Conceitos de Software

- Loader - Carregador
 - Carga do programa na memória para execução
 - Absoluto x Relocável
- Debugger - Depurador
 - Pesquisa de erros de lógica
 - Rastreamento

Conceitos de Software

- Linguagem de Controle
 - Comunicação Usuário com o SO
 - Caracter x Gráfica
 - Interpretador de Comandos - Shell
 - Arquivos de Comandos

Conceitos de Software

- Linguagem de Máquina
 - Instruções do Processador
 - Acesso aos Registradores / Memória
 - Modos de endereçamento
 - Tipos de dados

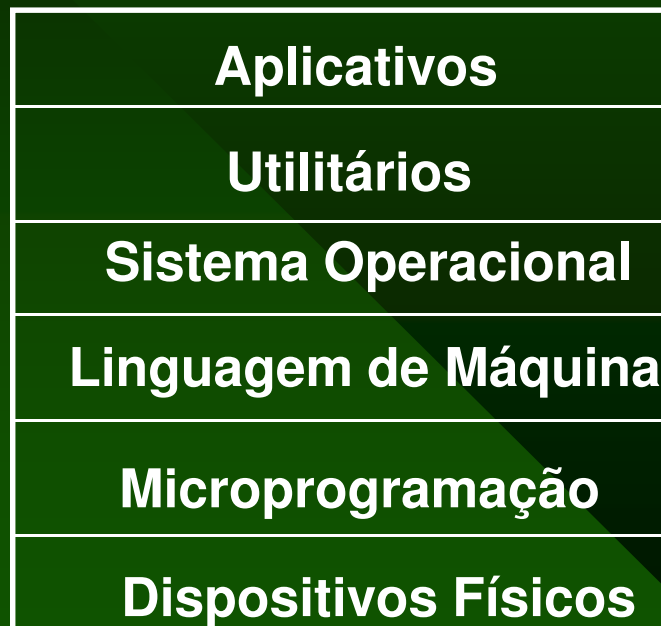
Conceitos de Software

- Linguagem de Máquina
 - Programa em linguagem de máquina
 - Totalmente codificado em binário
 - Diretamente processado pela CPU
 - Não requer tradução ou relocação
 - Não pode ser executado em outros processadores

Conceitos de Software

- Microprogramação
 - Cada instrução em linguagem de máquina tem um microprograma associado
 - O microprograma é formado por várias microinstruções
 - Microinstruções são executadas pelo hardware
 - Arquitetura CISC
 - Processadores microprogramáveis
 - Aceitam novas instruções - novos microprogramas

Máquina de Níveis



Histórico de SOs

- Primeira Geração (1945 - 1955)
 - Primeiros computadores
 - ENIAC (Electronic Numerical Integrator and Computer)
 - Realização de cálculos balísticos
 - Utilização de válvulas
 - ENIAC
 - 18.000 válvulas
 - Programação em linguagem de máquina

Histórico de SOs

- Primeira Geração (1945 - 1955)
 - Ausência de sistema operacional
 - EDVAC (Electronic Discrete Variable Automatic Computer)
 - Usuários
 - Universidades e órgãos militares
 - UNIVAC
 - Censo americano de 1950

Histórico de SOs

- Segunda Geração (1956 - 1965)
 - Criação dos transistores
 - Aumento na velocidade do processamento
 - Dimensão dos computadores
 - Criação das memórias magnéticas
 - Acesso mais rápido aos dados
 - Primeiras linguagens de programação:
 - Assembly e FORTRAN

Histórico de SOs

- Segunda Geração (1956 - 1965)
 - Cartão perfurado
 - Processamento *batch*
 - Processamento de lote de programas
 - SO com conjunto de rotinas para operações de Entrada e Saída (IOCS)
 - Conceito:
 - Independência dos dispositivos

Histórico de SOs

- Terceira Geração (1966 - 1980)
 - Monitor de vídeo e do teclado
 - Interação
 - *Time Sharing*
 - Surgimento do sistema operacional UNIX (linguagem C)
 - Primeiros microcomputadores

Histórico de SOs

- Terceira Geração (1966 - 1980)
 - Introdução dos Circuitos Integrados
 - Menor custo e dimensão
 - Performance
 - Multiprogramação
 - Compartilhamento da memória principal
 - Primitivas com bloqueio
 - Sinais e interrupções

Histórico de SOs

- Quarta Geração (1981 - 1990)
 - Aperfeiçoamento dos circuitos integrados
 - Surgimento dos PC's e do DOS
 - Substituição do CP/M
 - Estações de trabalho (monousuárias)
 - Multitarefa
 - Multiprocessadores
 - Sistemas operacionais de rede e distribuídos.

Histórico de SOs

- Quinta Geração (1991 - 2007)
 - Arquitetura cliente-servidor
 - Processamento distribuído
 - Multiprocessadores não convencionais
 - Linguagem natural;
 - Segurança, gerência e desempenho do SO e da rede
 - Consolidação dos sistemas de interfaces gráficas
 - Interação com usuários mais flexível

Tipos de Sistemas Operacionais

- Monoprogramáveis / Monotarefa
- Multiprogramáveis / Multitarefa
- Multiprocessados

Tipos de Sistemas Operacionais

- Sistemas Monoprogramáveis / Monotarefa
 - Todos recursos do sistema dedicados a uma tarefa
 - Execução de programas seqüencialmente
 - Usado nos primeiros computadores de grande porte
 - Usado nos primeiros computadores de pequeno porte
 - Sistemas Monusuário

Tipos de Sistemas Operacionais

- Sistemas Monoprogramáveis / Monotarefa
 - Sub-utilização dos recursos do sistema
 - Processador X Operações de E/S
 - Memória
 - Dispositivos de E/S
 - Implementação simples
 - Sem recursos de proteção

Tipos de Sistemas Operacionais

- Sistemas Multiprogramáveis / Multitarefa
 - Recursos do sistema compartilhados por diversas tarefas
 - Execução de programas concorrentemente
 - Aumento da produtividade
 - Redução de custos
 - Suporte a Sistemas Multiusuário
 - Compartilhamento na utilização dos recursos do sistema
 - Processador X Operações de E/S
 - Memória
 - Dispositivos de E/S

Tipos de Sistemas Operacionais

- Sistemas Multiprogramáveis / Multitarefa
 - Implementação complexa
 - Gerenciamento dos acessos concorrentes aos recursos
 - Recursos de proteção
 - Sistemas Monousuário X Multiusuário
 - Mainframes
 - Computadores Pessoais e Estações de Trabalho

Tipos de Sistemas Operacionais

- Sistemas Multiprogramáveis / Multitarefa
 - Sistemas de Tempo Real
 - Semelhante aos Sistemas de Tempo Compartilhado
 - Limites rígidos para tempo de resposta
 - Sem fatia de tempo
 - Níveis de prioridade
 - Controle de processos (Indústrias, Tráfego Aéreo)

Tipos de Sistemas Operacionais

- Multitarefa
 - Colaborativa
 - Windows 95, 98
 - Não existe fatia de tempo
 - Preemptiva
 - Windows NT, XP, Linux, Mac OS, ...
 - Existe fatia de tempo
 - Preempção

Tipos de Sistemas Operacionais

- Sistemas Multiprocessados
 - Sistemas com mais de uma CPU interligada
 - Execução simultânea de programas
 - Supre dificuldade no desenvolvimento de processadores mais rápidos
 - Ideal para sistemas que necessitam uso intensivo de CPU
 - Processamento científico

Tipos de Sistemas Operacionais

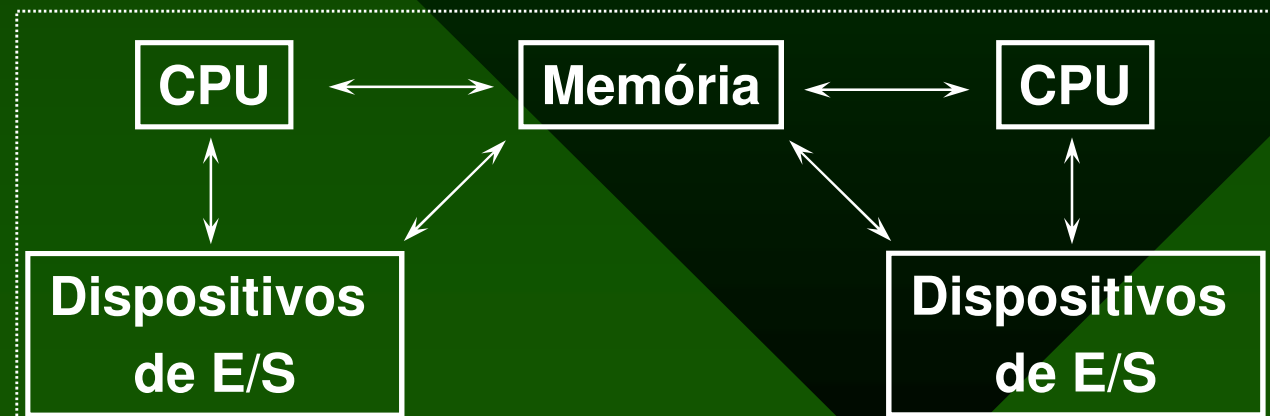
- Sistemas Multiprocessados
 - Características
 - Multiprogramação
 - Aplicada a cada processador
 - Escalabilidade
 - Aumento da capacidade computacional
 - Reconfiguração
 - Tolerância à falha em algum processador
 - Balanceamento
 - Distribuição de carga de processamento

Tipos de Sistemas Operacionais

- Sistemas Multiprocessados
 - Classificação
 - Em função:
 - da forma de comunicação entre CPUs
 - do grau de compartilhamento da memória e E/S
 - Sistemas Fortemente Acoplados
 - Processadores com múltiplos núcleos, por exemplo.
 - Sistemas Fracamente Acoplados
 - *Server Cluster*, por exemplo.

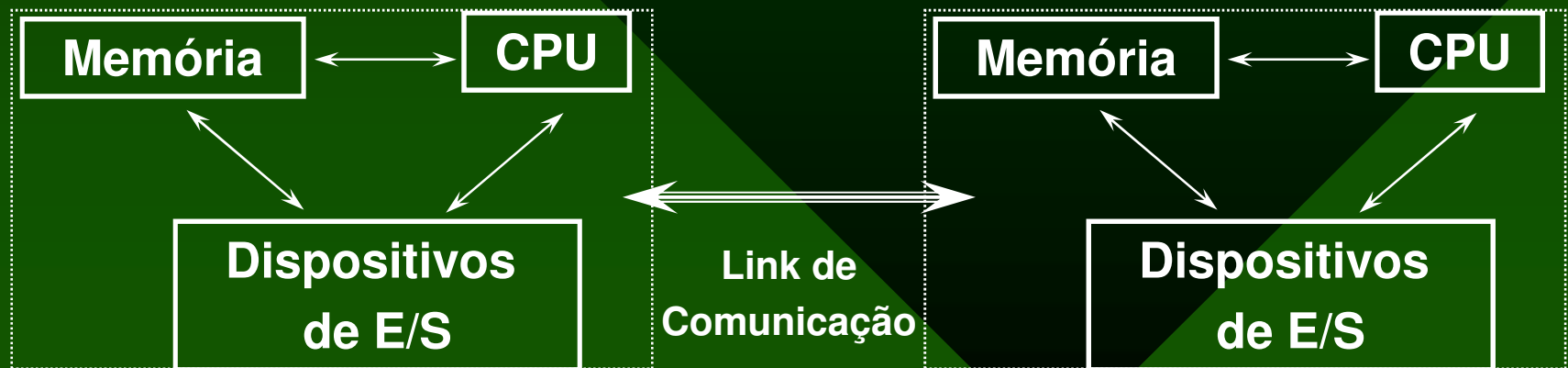
Tipos de Sistemas Operacionais

- **Sistemas Multiprocessados**
 - **Sistemas Fortemente Acoplados**
 - Processadores compartilham um única memória
 - Espaço de Endereçamento Único
 - Único SO



Tipos de Sistemas Operacionais

- Sistemas Multiprocessados
 - Sistemas Fracamente Acoplados
 - » Sistemas de Computação independentes, mas conectados (multicomputadores)
 - u Processamento Distribuído
 - u SO de Rede (SOR) X SO Distribuído (SOD)



Tipos de Sistemas Operacionais

- Multiprocessamento
 - Computadores vistos originalmente como máquinas seqüenciais
 - Execução seqüencial das instruções do programa
 - Sistemas Multiprocessados
 - Paralelismo - Simultaneidade
 - Execução de várias tarefas ou sub-tarefas

Definição de Processo

- Um programa em execução
- Não é o mesmo que Programa (entidade estática)
 - Programa é o código executável
 - Processo é o código executando
- Entidade dinâmica
 - Ex.: A execução de *prog.exe* 3 vezes gera 3 processos distintos do mesmo programa

Ambiente de um Processo

- Todo processo precisa ter:
 - Seção de texto
 - Código executável
 - Seção de dados
 - Variáveis, estruturas
 - Pilha do processo
 - Parâmetros etc
 - Registradores, incluindo PC, SP, BP etc.

Conceitos

- Para que os processos executem em ambiente multiprogramado, existe a gerência de:
 - Compartilhamento da CPU, de memória, E/S etc
- Nomenclatura
 - Sistemas Batch
 - *Job*
 - Sistemas de Tempo Compartilhado
 - Tarefa (*on-line*)

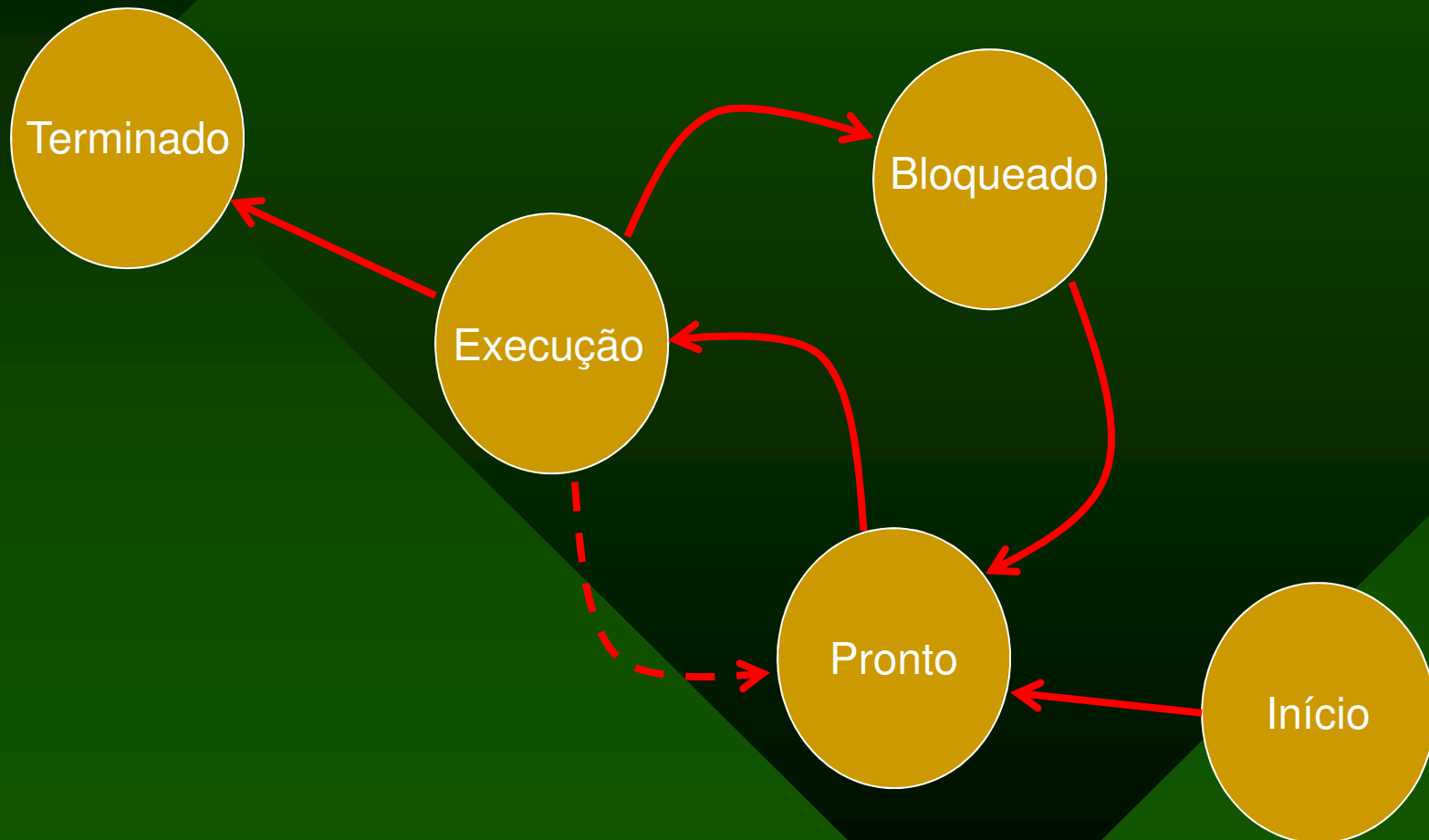
Estados do Processo

- Mudanças de estado durante a execução:
 - Iniciando
 - O processo está sendo criado
 - SO Aloca Memória, Contexto (BCP)
 - Ex.: O usuário *clica* num arquivo executável no Explorer ou executa via CMD
 - Executando
 - Instruções do processo estão sendo executadas
 - Bloqueado
 - O processo está esperando algum evento externo

Estados do Processo

- Mudanças de estado durante a execução:
 - Pronto
 - O processo está aguardando chance de ser executado no processador
 - Terminando
 - O processo está finalizando sua execução

Escalonamento de Processos



Controle de Processo

- Bloco de Controle de Processo (BCP)
 - Área de Memória alocada pelo SO para gerenciar processos
 - Mantém informações sobre o processo
 - Estado do processo
 - Registradores da CPU, incluindo o PC
 - Informações para escalonamento
 - Informações para gerenciamento de memória
 - Informações de contabilização
 - Informações sobre operações de E/S
 - Ponteiros para Arquivos, *Socket* etc

Troca de Contexto

- Preempção
 - Ação de retirar um processo da CPU a qualquer instante e restaurá-lo como se nada tivesse ocorrido
 - Suportado por Interrupções (como pelo CLOCK) e pelo BCP
 - Ex.: Preempção por tempo de Quantum
 - Preempção por I/O, Prioridade etc.
- Tempo da troca é considerado *overhead*
- Tempo depende de suporte de *hardware* e complexidade do SO

Escalonamento de Processos

- Manutenção de filas para controle dos processos
 - Fila de Processos – todos processos no sistema
 - Fila de Pronto – processos prontos
 - Filas de Bloqueado - processos aguardando E/S de dispositivo (interrupção) ou sinal
 - Múltiplas Filas – uma por dispositivo
 - Fila de Disco, Fila de CD, Fila de Teclado, de Rede
 - BCP efetua o encadeamento nas filas
 - Processos passam por várias filas

Escalonadores

- Classificação de processos:
 - I/O-BOUND – Intensivamente consumidor de E/S
 - Passa mais tempo fazendo operações de E/S do que utilizando a CPU
 - Processos Comerciais
 - CPU-BOUND – Intensivamente consumidor de CPU
 - Passa mais tempo efetuando cálculos do que E/S
 - Processos Científicos/Matemáticos
 - Híbrido

Escalonadores

- Combinação adequada de processos CPU e I/O BOUND melhora o desempenho e flexibilidade do sistema
 - Processo interativo (Ex.: Internet Explorer)
 - Tipicamente I/O Bound

Algoritmo de Escalonamento de CPU

- Algoritmo do S.O. que determina qual o próximo processo a ocupar a CPU
 - Executado quando ocorre estouro de Quantum ou interrupção do processo (I/O, Evento, Sinal etc.) ou o processo acaba;
- Critérios mudam com características dos Processos
 - *Batch, CPU Bound, I/O Bound, Interativos*

Metas do Escalonamento

- **Eficiência**
 - Manter a CPU ocupada 100% do tempo
- **Throughput**
 - Maximizar o número de processos (tarefas, jobs) executados em um dado intervalo de tempo
- **Turnaround**
 - Minimizar o tempo de um processo no sistema, desde seu início até o término
 - Tempo médio de execução
 - Fundamental a processos Batch

Metas do Escalonamento

- **Igualdade**
 - Todo Processo tem direito de ocupar a CPU
- **Tempo de resposta**
 - Minimizar o tempo decorrido entre a submissão de um pedido e a resposta produzida num processo interativo

Conflito entra Metas

- Atender a uma meta pode prejudicar outra
 - Qualquer algoritmo de escalonamento favorecerá um tipo de processo (*CPU Bound, I/O Bound, Tempo Real, etc*) em detrimento de outros
 - O propósito precisa ser geral

Tipos de Escalonamento

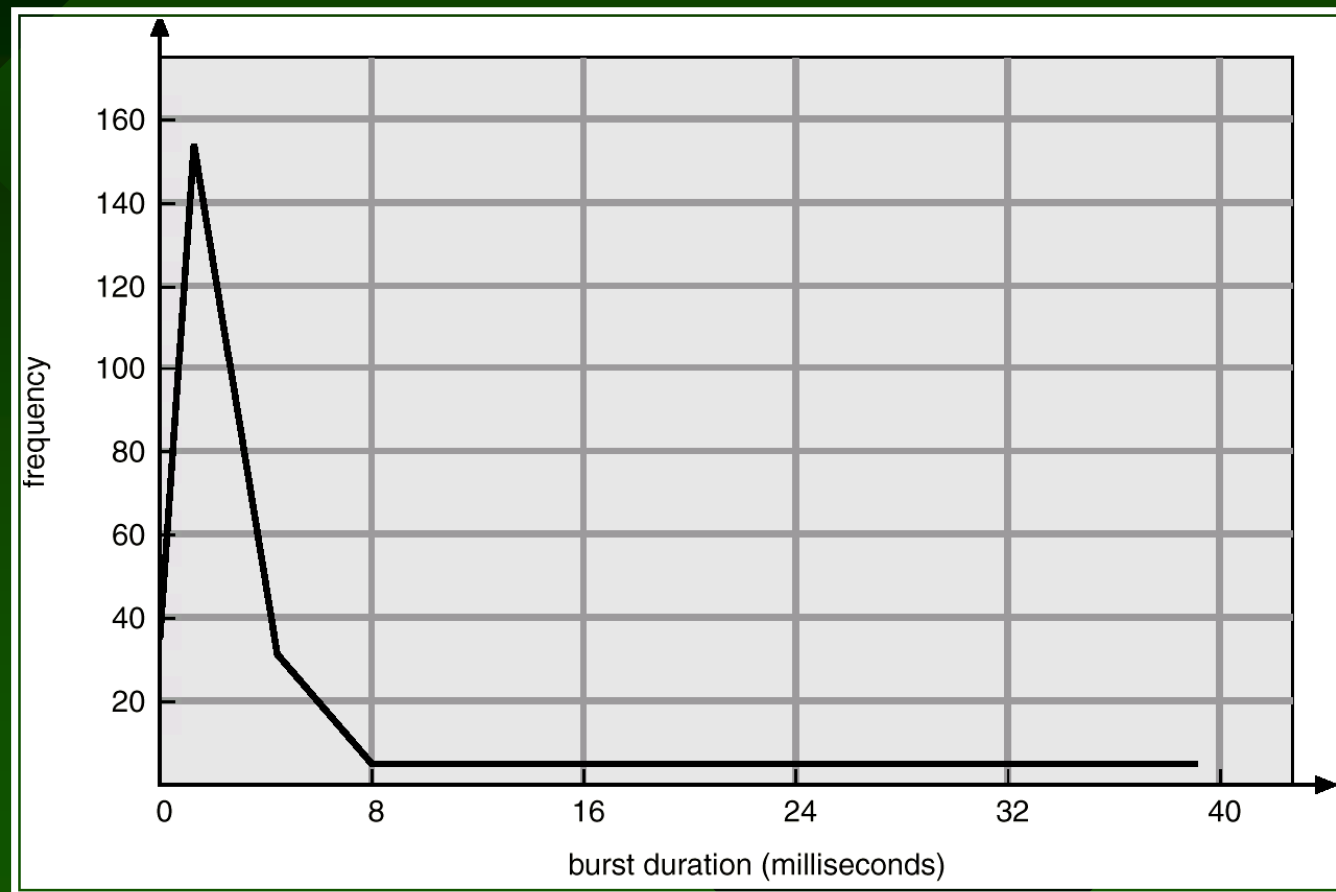
– Escalonamento não-preemptivo

- Escalonamento Cooperativo;
- Processo mantém a CPU até terminar ou E/S;
- Não requer recursos especiais de hardware
 - Não existe Quantum (devolução voluntária do controle ao S.O.)
- Usado até o Windows 3.x.

– Escalonamento preemptivo

- Requer temporizador na CPU (fatia de *quantum* ou Uso do *clock*);
- Requer suporte do SO para coordenar acesso a dados compartilhados de forma consistente (proteção).

Freqüência de processos por duração de surto de CPU



Escalonamento FIFO

- *First Come First Served* (FCFS, FIFO, PEPS)
 - Não preemptivo

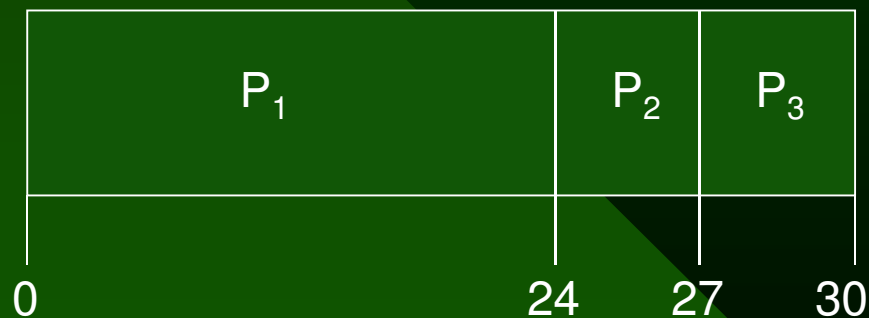
Processo	Início	Duração (ut)
P ₁	0	24
P ₂	0	3
P ₃	0	3

Escalonamento FIFO

- Ordem de chegada dos processos:

P_1, P_2, P_3

- Diagrama de Gantt



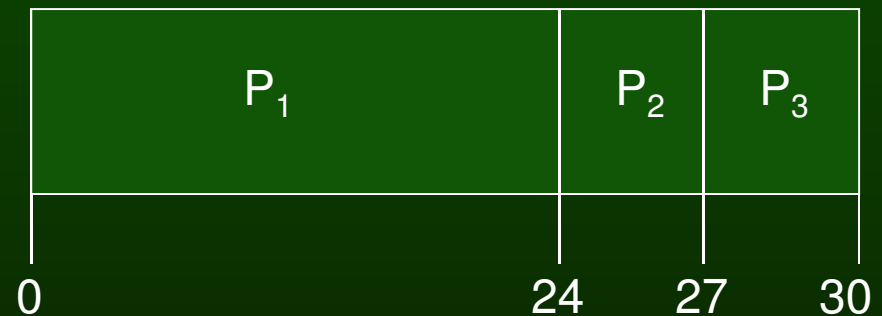
Escalonamento FIFO

- Tempos de espera

$$P_1 = 0$$

$$P_2 = 24$$

$$P_3 = 27$$



$$\text{Throughput} = 0,1 \text{ (3/30)}$$

Dica: Tempo de Espera é o tempo que o processo passa no estado de Pronto.

- Tempo médio de espera

$$(0 + 24 + 27) / 3 = 17$$

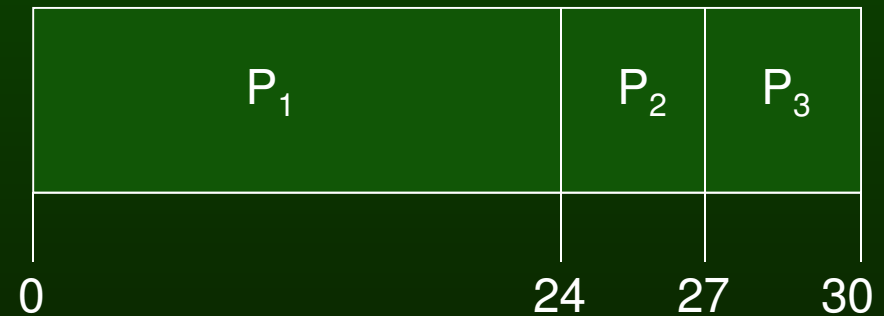
Escalonamento FIFO

- Tempos de saída

$$P_1 = 24$$

$$P_2 = 27$$

$$P_3 = 30$$



- Tempo médio de saída

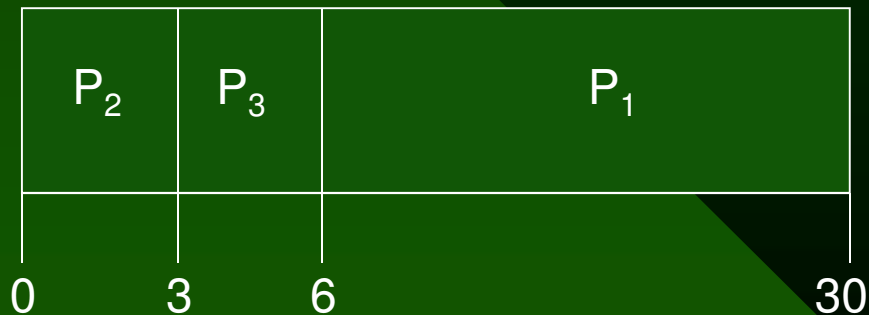
$$(24 + 27 + 30) / 3 = 27$$

Escalonamento FIFO

- Outra ordem de chegada

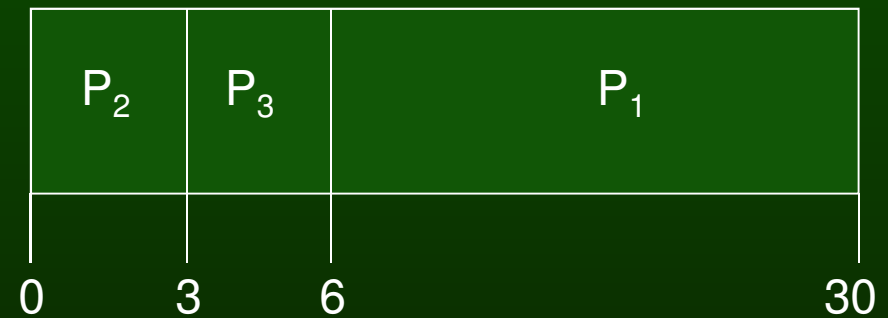
P_2, P_3, P_1

- Diagrama de Gantt



Escalonamento FIFO

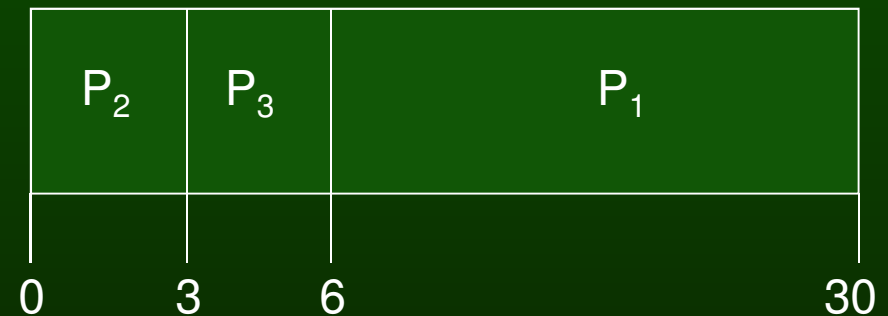
- FIFO ordenado (SJF / MPP)
 - Menor Processo Primeiro
 - Menor tempo de execução
- Tempos de espera
 - $TEP_1 = 6$; $TEP_2 = 0$; $TEP_3 = 3$
- Tempo médio de espera melhora
 - $(6 + 0 + 3) / 3 = 3$
- Tempo médio de espera não é mínimo
 - Pode variar muito (com os surtos de CPU)
- Efeito Comboio
 - Processos I/O bound esperam por CPU bound



Escalonamento FIFO

- Tempos de saída

$$P_1 = 30; \quad P_2 = 3; \quad P_3 = 6$$



- Tempo médio de saída melhora

$$(30 + 3 + 6) / 3 = 13$$

- Tempo médio de saída não é mínimo
 - Pode variar muito (com os surtos de CPU)

Escalonamento SJF

- *Shortest-Job-First* (Menor Job Primeiro)
 - Deveria ser “próximo surto de CPU menor primeiro”

Usado para Processos *batch*.

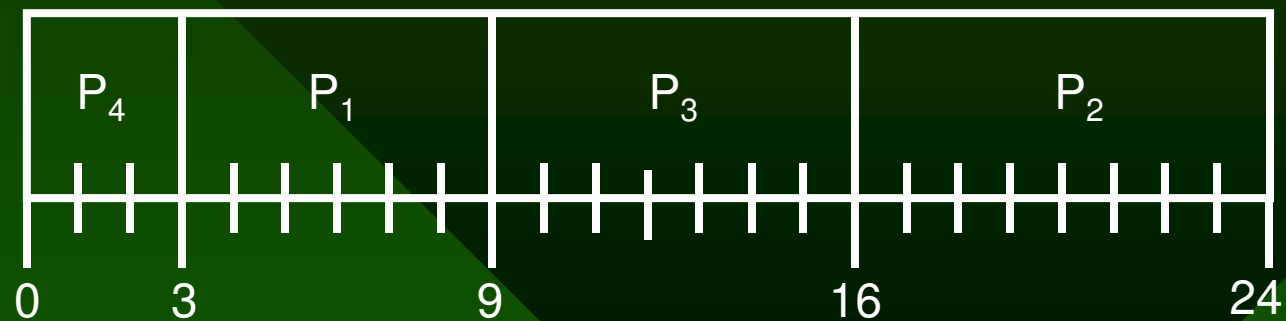
Sua execução diária permite determinar seu tempo total.

PID	Início	Duração de surto
P ₁	0	6
P ₂	0	8
P ₃	0	7
P ₄	0	3

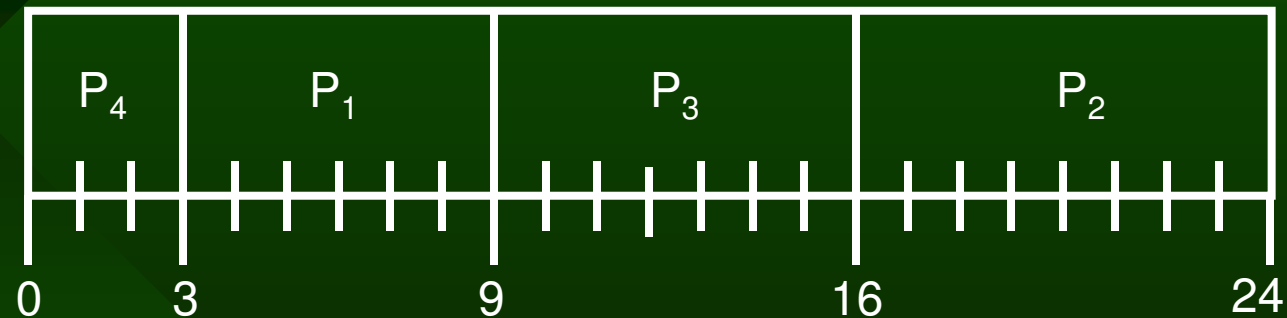
Escalonamento SJF

- Tempos de espera

$$P_1 = 3; \quad P_2 = 16; \quad P_3 = 9; \quad P_4 = 0$$



Escalonamento SJF



- **Tempo médio de espera melhora**

$$(3 + 16 + 9 + 0) / 4 = 7$$

Para FIFO, nesta situação, seria $10,25 = (0 + 6 + 14 + 21) / 4$

- **Tempo médio de espera é mínimo**

– Algoritmo considerado *ótimo*

Escalonamento SJF

- Problema: determinação exata da duração do próximo surto de CPU é impossível
 - SJF é usado para escalonamento de jobs em sistemas batch
 - Usuário especifica o tempo de CPU do job
 - Em escalonamento de CPU é usada estimativa
 - Baseada na duração dos surtos anteriores
 - Média exponencial

Preempção em SJF

- Não preemptivo
 - Processo usa CPU até completar surto
- Preemptivo
 - Novo processo pronto com surto previsto (T_A)
 - Tempo restante previsto para o processo em execução (T_B)
 - Se $T_A < T_B \Rightarrow$ preempção por prioridade
 - *Shortest-Remaining-Time-First (SRTF)*

Preempção em SJF

Processo	Instante de chegada	Duração de surto
P_1	0	7
P_2	2	4
P_3	4	1
P_4	5	4

Preempção em SJF

- SJF não preemptivo

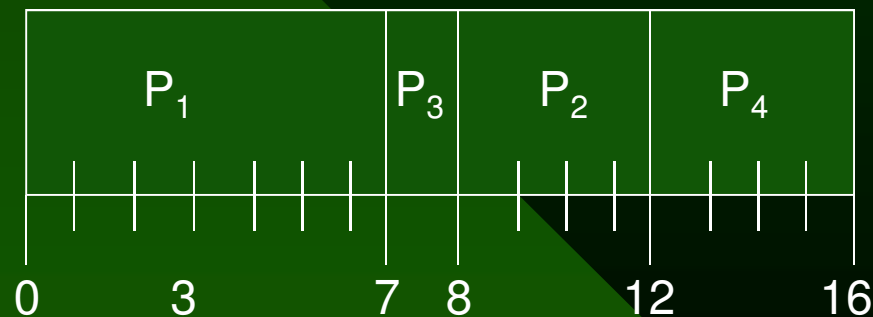
- Tempo de espera médio = $(0 + 6 + 3 + 7) / 4 = 4$

- $TEP_1 = 0$

- $TEP_2 = 6 \quad (8 - 2)$

- $TEP_3 = 3$

- $TEP_4 = 7$



Preempção em SJF

- SJF não preemptivo

- Tempo de saída médio = $(7 + 10 + 4 + 11) / 4 = 8$

- $TSP_1 = 7$

- $TSP_2 = 10$

- $TSP_3 = 4$

- $TSP_4 = 11$

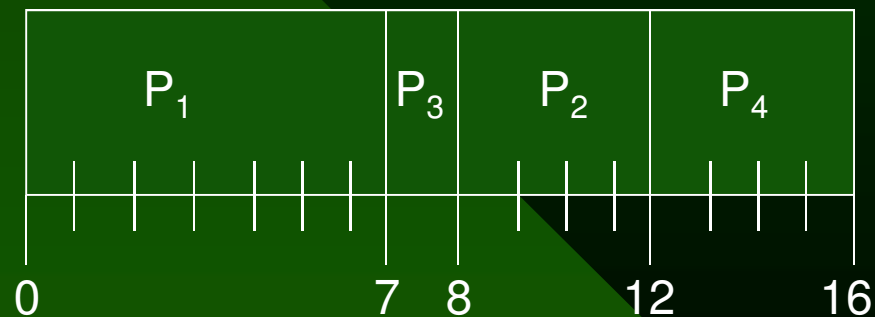


Tabela de Estados

Tempo	PR	EX	TER
0	P1 (7)		
0		P1 (7)	
2	P2 (4)	P1 (5)	
2	P1(5)	P2(4)	
4	P3(1), P1(5)	P2(2)	
4	P2(2), P1(5)	P3(1)	
5	P2(2), P4(4), P1(5)		P3
5	P4(4), P1(5)	P2(2)	
7	P4(4), P1(5)		P2
7	P1(5)	P4(4)	
11	P1(5)		P4
11		P1	
16			P1

TP2 < TP1 => Preempção

TP3 < TP2 => Preempção

Escalonador por Término de P3

Escalonamento *Round Robin*

- Round-Robin (revezamento circular)
 - Sistema Preemptivo
 - Interrupção do Clock (existe Quantum)
 - Tempo de espera médio é longo
 - Tempo de saída maior que SJF
 - Tempo de resposta melhor que SJF

Escalonamento *Round Robin*

- Preemptivo
 - Quantum de tempo (10 ~ 100 ms)
 - Necessita temporizador
 - Fila circular de processos prontos
 - Com quantum q e $n+1$ processos prontos:
 - Tempo máximo de espera: $n * q$

Escalonamento *Round Robin*

- Com quantum q e $n+1$ processos prontos:
 - Tempo máximo de espera: $n*q$
- Suponha uma fila de pronto com 101 processos, Quantum de 100 ms
- Um processo interativo executa, faz uma requisição, vai para bloqueado e de lá para o fim da fila
 - Quando a resposta será entregue ao usuário do processo interativo?

Escalonamento por Prioridade

- Cada processo tem uma prioridade
 - Número inteiro dentro de limites
 - Faixas 0 a 7 ou 0 a 4095
 - Menor (ou maior) número \Rightarrow maior prioridade
 - Empate \Rightarrow FCFS
 - SJF é um caso especial de prioridade

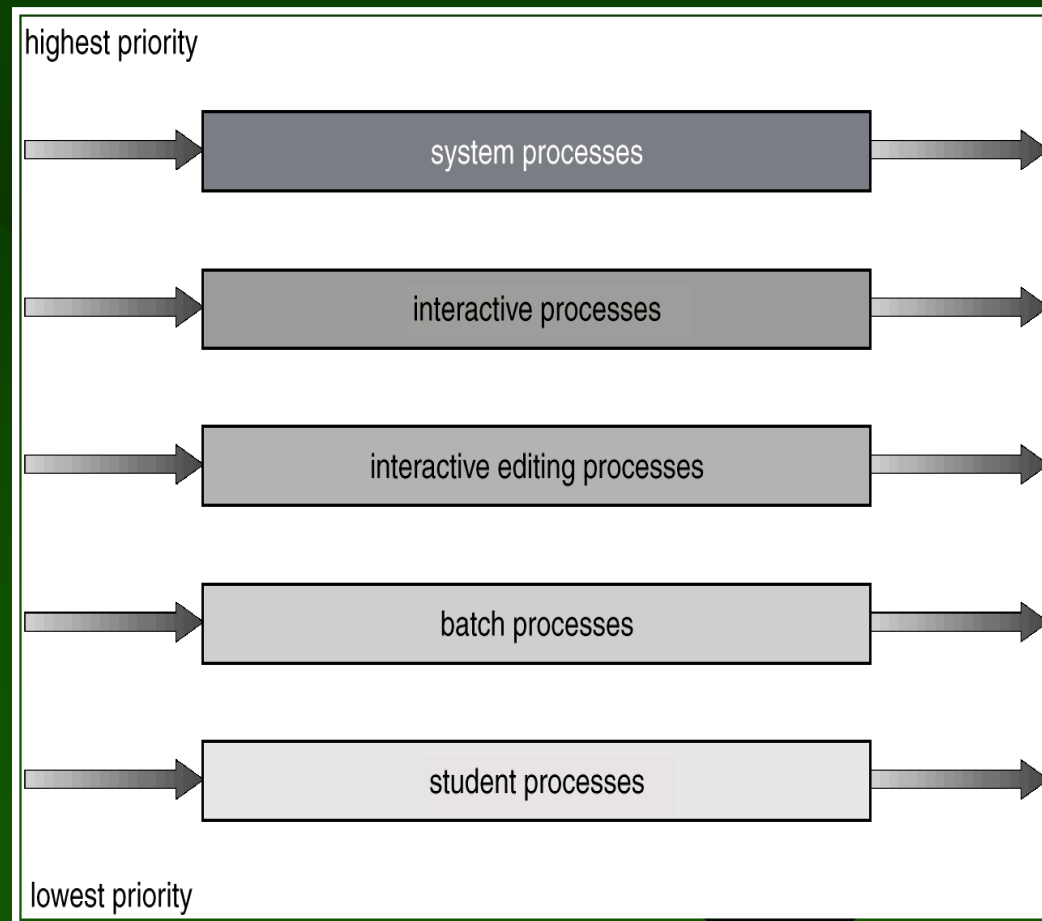
Escalonamento por Prioridade

- Prioridade definida interna ou externamente
- Preemptivo ou não preemptivo
- Starvation – Estagnação
 - Bloqueio por tempo indefinido
 - Solução: *aging* (envelhecimento)

Escalonamento por Múltiplas Filas

- Escalonamento preemptivo entre filas
 - Prioridade fixa: só atende filas menos prioritárias se as demais estiverem vazias
 - *Time slice* 80% para *foreground* com RR e 20% para *background* com FIFO

Escalonamento por Múltiplas Filas



Escalonamento por Múltiplas Filas

- Filas caracterizadas pelos surtos de CPU dos processos
 - *I/O bound* e interativos com mais prioridade
 - Passam a maior parte do tempo Bloqueados
 - Processos podem mudar de fila
 - *Aging* pode ser facilmente implementado
- Algoritmo preemptivo

Escalonamento com Múltiplos Processadores

- Escalonamento de CPU mais complexo
 - Existem sistemas com barramento de E/S privativo de determinado processador
- Várias filas de processos prontos
 - Possibilidade de desperdício de recursos

Escalonamento com Múltiplos Processadores

- Única fila de processos prontos
 - *Symmetric Multiprocessing (SMP)*
 - Cada processador faz seu escalonamento
 - Compartilhamento de estruturas de dados do SO
 - Sincronização
 - *Assymmetric Multiprocessing*
 - Escalonamento no processador mestre