

Universidade de São Paulo  
 Instituto de Ciências Matemáticas e de Computação  
 Departamento de Sistemas de Computação

## SSC 140 - SISTEMAS OPERACIONAIS I

Turmas A e B

### Aula 23 – Sistema Operacional Linux

Profa. Sarita Mazzini Bruschi

## Roteiro

- História
- Características Gerais
- Kernel
- Sistema Operacional GNU/Linux
  - Gerenciamento de Processos
  - Gerenciamento de Memória
  - Gerenciamento de E/S
- Linux e o Mercado de Trabalho
- Certificações

## História

- 1970
  - MIT + Bell Labs + General Electric = MULTICS (*multiplexed information and computing service*)
  - Com o fracassado do projeto MULTICS, a Bell manteve o pesquisador Ken Thompson, que escreveu um MULTICS mais enxuto, o qual foi apelidado por Brian Kernighan de UNICS (*uniplexed information and computing service*), e posteriormente mudou a ortografia para **UNIX**
- 1978
  - Dennis Ritchie aderiu ao projeto, onde o Unix foi reescrito em linguagem C com 10.000 linhas de código, tornando-se muito complexo (Unix PDP-11);

## História

- Andrew Tanenbaum
  - Desenvolve o **Minix**, um sistema com as características parecidas com as do Unix.
  - Objetivo: Estimular o aprendizado de sistemas operacionais. O código é reduzido e mais limitado;
- 1991
  - **Linus Torvalds** precisava utilizar o Unix para trabalhos acadêmicos, porém a disponibilidade de acesso era limitada, restando então a alternativa de se ter o Unix em casa
- Inspirado no **Minix**
  - Linus cria então um SO com um *kernel* mais poderoso, compatível com o Unix, e com código aberto: nasce o **Linux**, em sua primeira versão;

## História

- Distribuição do Linux
  - **O kernel** é o básico do sistema operacional Linux é o 2.6
  - A versão 2.4 do *kernel* ficou sob responsabilidade de um **brasileiro**, Marcelo Tossati. Período: 2001-2006
    - O fato do controle do núcleo ser centralizado faz com que as atualizações lançadas sejam estáveis;
  - Sob licença GPL (*GNU Public License*);

## História

- Empresas e Instituições independentes cuidam das distribuições do Linux.
  - Instalação
  - Conjunto de aplicativos
  - Suporte
  - Manuais
  - Preço
- Distribuições: Red Hat (Fedora, Mandriva), SuSE, Debian (Ubuntu), Slackware, e muitas outras

## Características Gerais

- ▣ Desenvolvido voluntariamente por programadores de todo o mundo;
- ▣ Multiusuário;
- ▣ Obedece aos padrões POSIX, ANSI, ISO, IETF e W3C;
- ▣ Executa em praticamente qualquer tipo de processador (depende da interface), com arquiteturas diferentes

## Kernel monolítico

- ▣ O kernel é projetado para ser um conjunto de módulos individualmente carregáveis
- ▣ Cada módulo possui uma interface bem definida que mostra como suas funcionalidades podem ser chamadas e como seus dados podem ser acessados por outros módulos
- ▣ De modo oposto, a interface também mostra as funcionalidades e dados dos outros módulos que podem ser usados por esse módulo

## Kernel monolítico

- ▣ Poucos módulos são carregados no momento do *boot*, sendo que novos módulos necessários são carregados dinamicamente
- ▣ No entanto, eles têm que ser integrados com os módulos que já estão carregados na memória, de modo que os módulos possam funcionar coletivamente como um kernel monolítico

## Kernel 2.6

- ▣ O kernel 2.6 eliminou problemas do kernel 2.5, dentre eles:
  - Tornar o sistema mais interativo
    - ▣ O kernel 2.5 era não-preemptivo no sentido de tarefas de altas se baixas prioridades, ou seja, se uma tarefa do kernel de baixa prioridade estava sendo executada, a de alta prioridade do kernel tinha que esperar
  - Suportar sistemas embarcados
    - ▣ Suporte a arquiteturas que não possuem MMU

## Arquitetura do Kernel

- ▣ O linux é composto de 6 subsistemas primários:
  - Gerenciamento de processo
  - Comunicação interprocesso
  - Gerenciamento de memória
  - Gerenciamento de sistema de arquivo
  - Gerenciamento de E/S
  - Gerenciamento de rede

## Gerenciamento de Processos

- ▣ No linux, tanto processos quanto *threads* são denominados **tarefas**
- ▣ Um processo é representado por uma estrutura de dados especial chamada de descritor de processos (*task\_struct*):
  - Mantém informações de alto e baixo nível, desde cópia de registradores de hardware até o *i-node* do diretório de trabalho para o processo (BCP);
    - ▣ Ficam armazenadas em uma tabela de processos (reside em uma página de memória reservada), de tamanho finito (*default* → 512K);

## Gerenciamento de Processos

- ❑ Assim que um processo é criado, recebe uma identificação (*pid*);
- ❑ Permite manipulação de *threads*;
  - Escalonamento é baseado em *threads*;
- ❑ Processos são divididos em três classes:
  - Processos interativos;
  - Processos *batch*;
  - Processos em tempo real;
- ❑ Cada classe pode ainda ser dividida em:
  - *I/O bound* (privilegiados pelo escalonador);
  - *CPU bound*;

## Gerenciamento de Processos

- ❑ Estados dos Processos no Linux:
  - **Executando**: em execução na CPU ou pronto para ser executado;
  - **Esperando (adormecido)**: equivalente a processo bloqueado, aguardando algum evento ou recurso;
  - **Parado**: parado devido a ter recebido um sinal de interrupção;
  - **Zombie**: é um processo já encerrado, mas que por algum motivo ainda possui uma estrutura alocada;
    - ❑ Ocorre quando o processo pai ainda não leu seu código de saída, permitindo que o *kernel* "limpe" a estrutura alocada;

## Gerenciamento de Processos Escalonamento

- ❑ Escalonamento baseado em prioridade, sendo que essa prioridade afeta o tamanho de seu período de tempo e a ordem em que ela executa
- ❑ O escalonador reconhece 40 níveis distintos de prioridade que vão de -20 a 19, sendo que a -20 é a de mais alta prioridade
- ❑ Se o processo for interrompido por E/S, o escalonador eleva dinamicamente a prioridade da tarefa, diminuindo o valor da prioridade estática

## Gerenciamento de Processos Escalonamento

- ❑ Políticas:
  - **SCHED\_FIFO**:
    - ❑ Para processos em tempo real;
    - ❑ Prioridade estática;
    - ❑ Escalonador é chamado quando:
      - Processo é criado e sua prioridade é maior que a do corrente;
      - Processo libera espontaneamente o processador para processos com mesma prioridade;
      - Quando um processo sendo executado termina ou é bloqueado;

## Gerenciamento de Processos Escalonamento

- **SCHED\_RR**:
  - ❑ Para processos em tempo real;
  - ❑ Prioridade com *quantum* para cada processo;
  - ❑ Escalonador é chamado quando:
    - Processo terminou seu *quantum*;
    - Processo com maior prioridade está pronto para ser executado;
    - Processo libera espontaneamente o processador para processos com mesma prioridade;
    - Quando um processo sendo executado termina ou é bloqueado;
- **SCHED\_OTHER**:
  - ❑ Para processos interativos e *batch*;
  - ❑ Prioridades dinâmicas com tempo compartilhado;

## Gerenciamento de Memória

- ❑ **LMM (Linux Memory Manager)**: provê mecanismos que permitem que a memória do sistema possa ser utilizada da melhor maneira possível;
- ❑ **SO** encarrega-se de fornecer para cada processo um espaço suficiente para que ele possa ser executado, garantindo proteção;

## Gerenciamento de Memória

### Modelo de memória para um processo:

- Parte baixa de memória:
  - Texto: código processo;
  - Dados não inicializados/inicializados: espaço de armazenamento necessário às variáveis alocadas estaticamente no processo;
- Parte alta de memória:
  - Pilha: espaço de memória necessário às variáveis locais, para passagem de parâmetros e para salvar e restaurar endereços de retorno;
- HEAP: possibilita alocação dinâmica de memória; está entre a pilha e os dados;

## Gerenciamento de Memória

- Espaço de Endereçamento disponível: 4Gb (32bits) → corresponde à memória virtual;
  - 3Gb para processos;
  - 1Gb para tabelas de páginas e outros dados do *kernel*;
- SO mantém na memória somente as partes de código, dados e pilha que um processo está utilizando em um determinado tempo;
  - Mantém apenas uma estrutura interna que descreve a memória virtual do processo, indicando o que está carregado na memória e o que está no disco;
    - Paginação por demanda;

## Gerenciamento de Memória

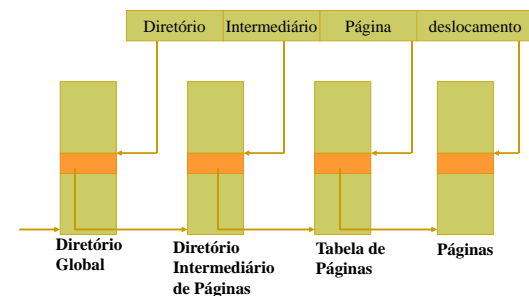
### Paginação

- Divide toda a memória disponível em páginas de tamanho fixo (4k p/ Pentium) e mantém uma tabela das páginas na memória;
  - A medida que os processos são executados, eles são carregados em páginas na memória, e a tabela é atualizada;
- Paginação em três níveis (portabilidade – 32 e 64bits):
  - Diretório global de páginas;
  - Diretório intermediário de páginas;
  - Tabela de páginas;

## Gerenciamento de Memória

### Paginação

#### Endereço Virtual



## Gerenciamento de Memória

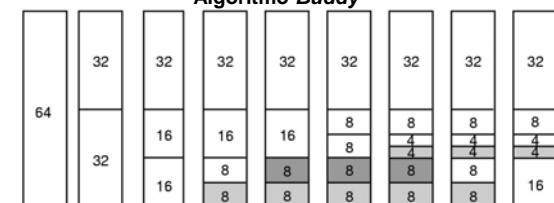
### Paginação

- Alocar e liberar páginas físicas: algoritmo *Buddy*;
- Controle de áreas de memória livres e alocadas é feito usando um vetor, no qual cada elemento é uma lista encadeada (*free-area*);
- Quando um processo precisa de informações que não estão presentes na memória ele causa uma falta de página e o SO é encarregado de buscar essa informação no disco (*Pager*);
  - Política para substituir páginas: **LRU** combinado com o algoritmo do relógio (ordem de endereço virtual);

## Gerenciamento de Memória

### Paginação

#### Algoritmo Buddy



- Inicialmente, a memória consiste de uma divisão contínua única: 64 páginas;

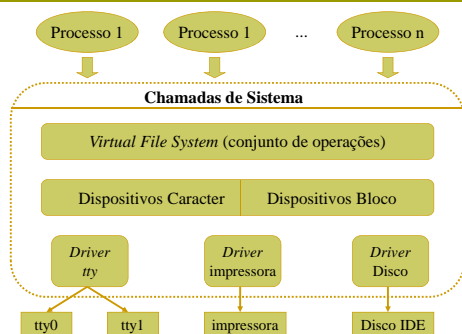
## Gerenciamento de E/S

- ❑ Requisições são realizadas por meio de chamadas de sistema (primitivas);
- ❑ Cada dispositivo é tratado como um arquivo especial → */dev*;
  - Tipo: caracter ou bloco;
  - Maior: identifica classe (1-255); mesmo *device driver*;
  - Menor: identifica um dispositivo específico;
  - Cada arquivo está associado a uma partição lógica do disco;
- ❑ Cada *driver* está associado a uma controladora;

## Gerenciamento de E/S

Nome	Tipo	Maior	Menor	Descrição
/dev/fd0	Bloco	2	0	Unidade de disquete 0
/dev/hda	Bloco	3	0	Disco IDE primário
/dev/hda1	Bloco	3	1	1ª Partição do disco IDE primário
/dev/hda2	Bloco	3	2	2ª Partição do disco IDE primário
/dev/lp1	Caracter	4	0	Impressora
/dev/console	Caracter	1	1	Teclado

## Gerenciamento de E/S



## Linux e o Mercado de Trabalho

- ❑ Linux tem se tornado um padrão nas empresas
- ❑ Substituindo/Integrando-se com sistemas caros e que não atendem determinadas necessidades
- ❑ Grandes empresas tem investido muito no Linux: Google, IBM, HP, etc.

## Linux e o Mercado de Trabalho

- ❑ Quem entende bem de sistemas Linux, seguramente terá um bom lugar no mercado de trabalho
- ❑ Por que?
  - Porque as empresas possuem diversas necessidades, dentre elas:
    - ❑ Migração e interligação de sistemas
    - ❑ Desenvolvimento de softwares para GNU/Linux

## Linux e o Mercado de Trabalho

- ❑ Ceagesp
- ❑ Band
- ❑ CELEPAR
- ❑ Casas Bahia
- ❑ Banco do Brasil
- ❑ Itaú
- ❑ Bradesco
- ❑ Embraer
- ❑ Metrô SP
- ❑ Sabesp

## Linux e o Mercado de Trabalho

- ❑ Casas Bahia
  - Maior rede varejista do país em setores de eletrodomésticos e eletroeletrônicos
  - 477 lojas em todo o Brasil
- ❑ Centro de tecnologia próprio
  - São Caetano do Sul – SP
  - Início em 2005
  - **Tudo** roda em LINUX (de estações de trabalho, terminais remotos e servidores)

## Linux e Mercado de Trabalho

- ❑ Vantagens segundo CEO das casas Bahia:
  - Economia na compra de computadores mais simples e baratos
  - Não pagamento de licenças
  - Melhor disponibilidade da infraestrutura de TI
  - Melhor desempenho com máquinas mais baratas e antigas
  - O **lucro** certamente também é maior

## Certificações

- ❑ Red Hat Certified Technician (RHCT)
  - A primeira certificação (conteúdo básico e resolução de problemas mais simples)
- ❑ Red Hat Certified Engineer (RHCE)
  - Instalação, configuração, etc.
- ❑ Red Hat Certified Architect (RHCA)
  - Design de infraestrutura de TI
- ❑ CLP – Novell Certified Linux Professional
- ❑ CLE – Novell Certified Linux Engineer

## Certificações

- ❑ Linux Professional Institute (LPI)
  - Dan York (Canadá)
- ❑ Características:
  - Neutro (não foca em distribuições e sim na essência do sistema operacional)
  - Baseado em princípios colaborativos
- ❑ Inclui 3 níveis

## Certificações

- ❑ LPIC Level 1
  - 101 – Gerenciamento de sistemas Linux
  - 102 – Gerenciamento de Redes
- ❑ LPIC Level 2 (A complexidade aumenta)
  - 201 – Gerenciamento de sistemas Linux
  - 202 – Gerenciamento de Redes
- ❑ LPIC Level 3 (Nível sênior)
  - 301 – Gerenciamento de projetos, migração Linux
  - 302 – Sólidos conhecimentos sobre protocolos
    - ❑ Foco em ambientes heterogêneos e de missão crítica

## Certificações

- ❑ LPIC Level 3 (Nível sênior)
  - 301
    - ❑ Conceito e arquitetura, Replicação de sites, design de diretório, migração de NIS para LDAP, integração com samba, tuning de performance, desenvolvimento de scripts
  - 302
    - ❑ Conceitos do protocolo SMB/CIFS, integração e configuração de diferentes clientes CIFS, administração avançada de recursos compartilhados, integração com kerberos, clustering

## Certificações

---

- ▣ Como e onde fazer as provas
  - <http://www.lpibrasil.com.br>
- ▣ Material para estudo
  - LPI Linux Certification in a Nutshell" – O'Reilly
  - Artigos sobre as provas de nível 1
    - ▣ <http://www.oreilly.com/catalog/lpicertnut2>
  - Foca Linux
    - ▣ <http://focalinux.cipsga.org.br/download-lpi.html>
  - Wiki Oficial do LPI nível 3
    - ▣ <https://group.lpi.org/cgi-bin/publicwiki/view/Examdev>
  - Lista de Discussão
    - ▣ <http://list.lpi.org/cgi-bin/mailman/listinfo/lpi-brasil>

## Link para um mapa interativo

---

- ▣ [http://www.makelinux.net/kernel\\_map#sd](http://www.makelinux.net/kernel_map#sd)