

PCS3446 – Sistemas Operacionais

Prof. João José Neto

Estudos de Casos

Aula 17 - UNIX

(+ LINUX, GNU)



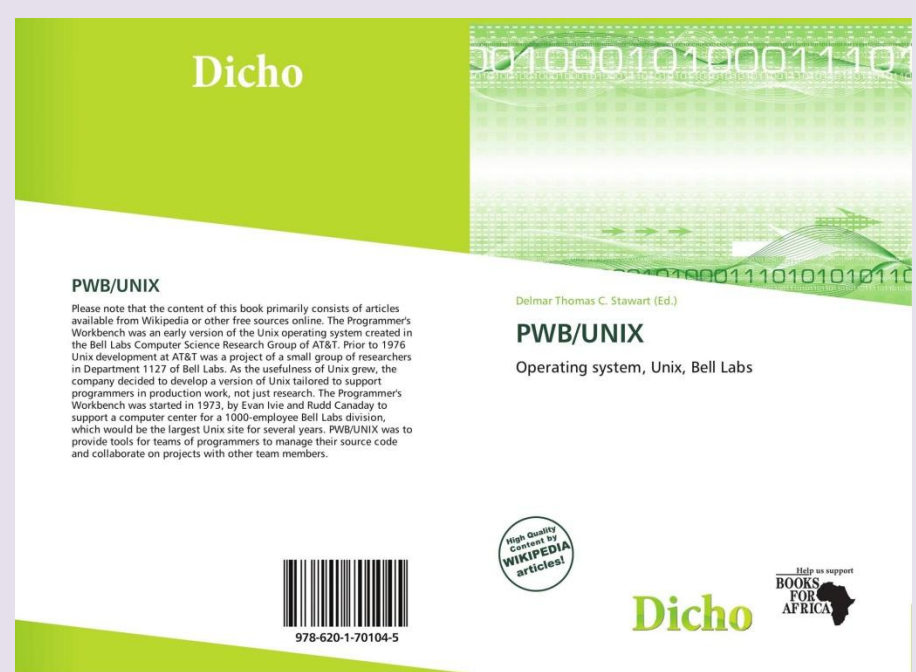
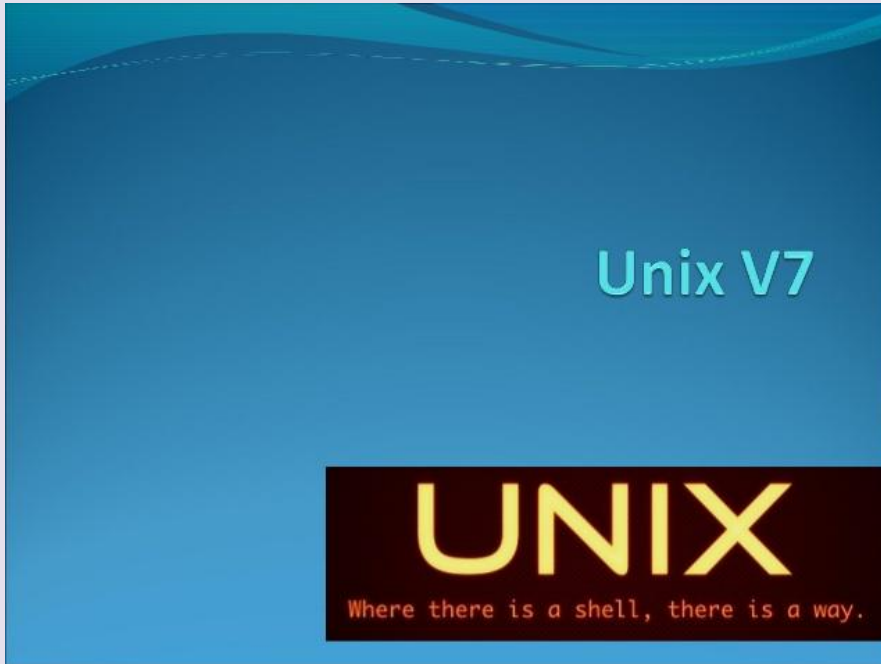
O SISTEMA OPERACIONAL UNIX

O Sistema Operacional UNIX

- Projetado como sistema voltado para o **desenvolvimento de programas**
- Linguagem de comandos **simples e poderosa**
- Sistema de arquivos **independente de dispositivo**

Versões diversas

- Há **versões** UNIX para **inúmeras máquinas**, entre as quais se destacaram:
 - Standard UNIX para a família DEC PDP-11
 - PWB/UNIX *Programmer's Workbench*
 - MIN-UNIX versão *downsized* do *Standard UNIX*
 - UNIX/V7 a versão 7 para DEC PDP 11/45 e 11/70



UNIX®

Sistemas DEC PDP-11 /45 e /70



PDP-11/45



PDP-11/70

Standard UNIX para a família DEC PDP-11

- Essa versão foi projetada para atender **centros de computação pequenos**, com uns 40 usuários.
- Dispunha dos seguintes recursos:
 - Sistema de arquivos hierárquico
 - Proteção
 - Independência de dispositivo
 - Programas em *foreground* e *background*
 - *Pipes*
 - Conjunto de programas e bibliotecas de sistema
 - Programas-fonte e documentação on-line

PWB/UNIX *Programmer's Workbench*

- Esta variante do UNIX disponibiliza:
 - Suporte para **programação em equipes grandes**
 - Entrada remota de jobs (RJE)
 - Sistema completo de editoração
 - *Source code control system* (SCCS) para
 - **Controle de versões** de software
 - **Documentação** de software

MIN-UNIX

- É uma versão *downsized* do *Standard* UNIX
- Voltado para **plataformas pequenas**
- Atende até 4 usuários
- Suporta até 13 processos concorrentes

UNIX/V7

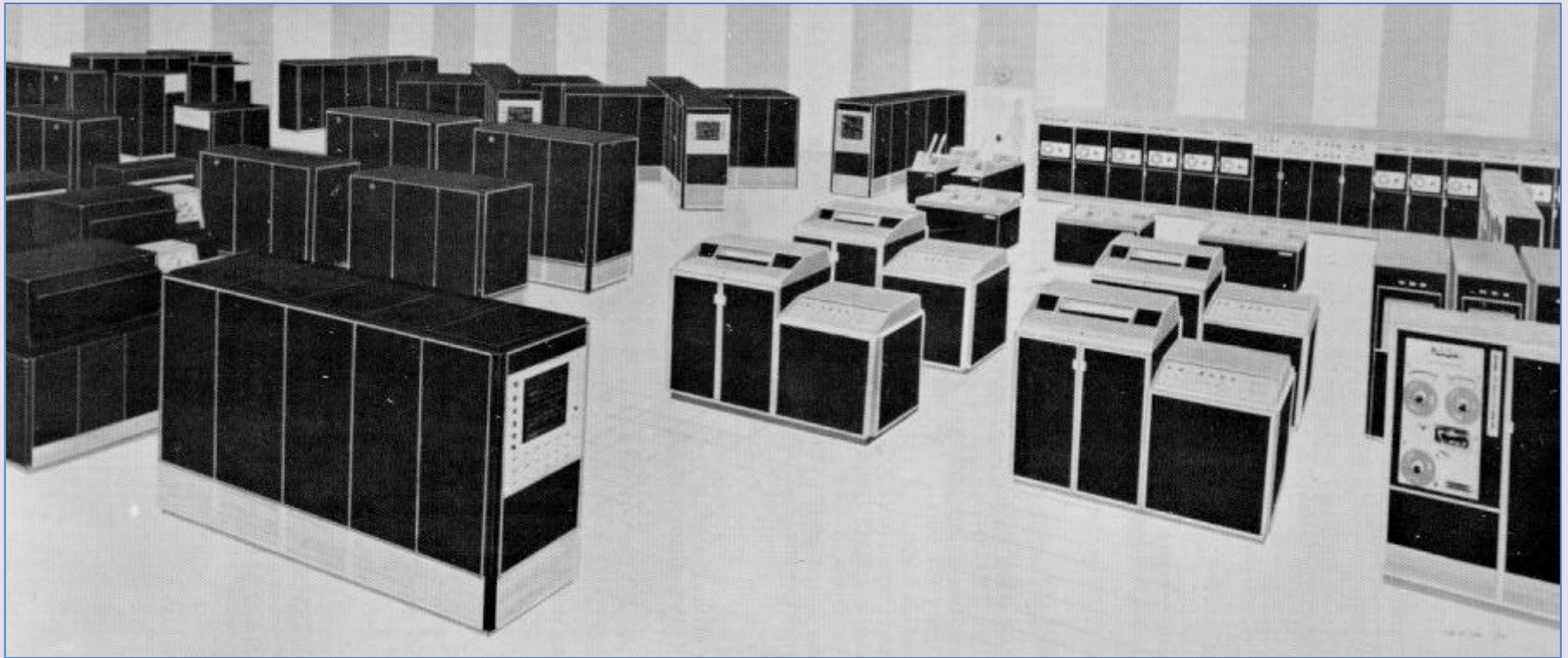
- A versão 7 foi ajustada para PDP 11/45 e 11/70
- Mais **portátil**
- Oferece linguagem C estendida
- Fornece apoio à edição de texto
- Apresenta um *shell* mais poderoso
- Facilita a transferência de arquivos entre máquinas

Histórico do Sistema Operacional UNIX

- O sistema **Multics** foi criado de **1965 a 1969** em conjunto pela Bell Labs, GE (Honeywell), e o projeto MAC do MIT
- **Originalmente**, foi projetado para ser executado em **grandes plataformas GE-645**

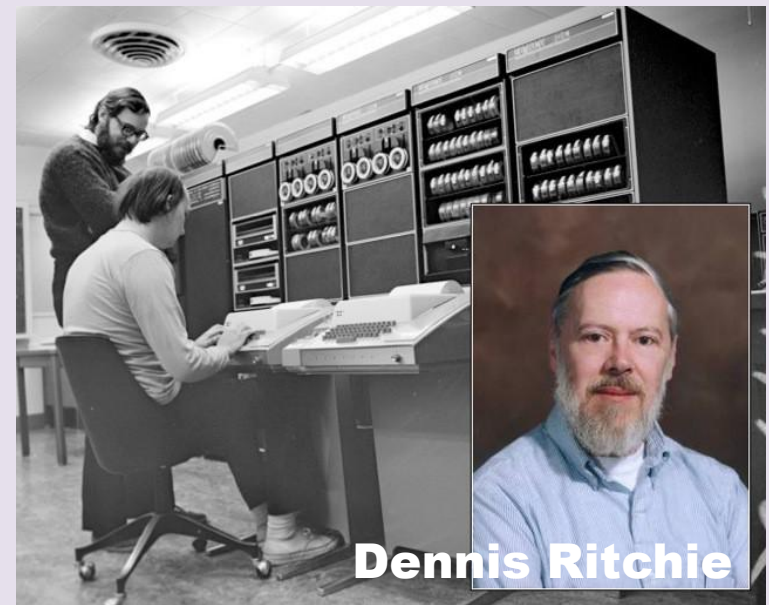


Sistema GE-645



The logo for Multics, featuring the word "multics" in a bold, blue, sans-serif font. A red dot is positioned above the letter 'i'.

- **Multics** era um sistema operacional **complexo e grande**, que daria origem ao UNIX.
- Era voltado para atender as necessidades de **computadores de grande porte**, com um grande número de usuários.
- Em 1969 a *Bell Labs* se retirou deste projeto.



- UNIX começou como um sistema de *time-sharing* para minicomputadores
- Alguns membros da *Bell Labs* resolveram criar o UNIX, para ser executado em plataformas **simples**, para **pesquisa e desenvolvimento de aplicações**, no sistema DEC PDP-7

Sistema DEC PDP-7



Linguagem de montagem do PDP-7

- Inicialmente o sistema UNIX foi desenvolvido na **linguagem de montagem** do computador PDP-7.
- O aspecto dos programas escritos nessa linguagem era semelhante ao do fragmento de código ao lado.

```
THIS IS A SAMPLE PROGRAM
GO,      LAS
          SPA      :CMA
          JMP      GO
          DAC      #CNTSET
          LAC      (1
          DAC      #BIT
          CLL
LOOP,    LAC      CNTSET
          DAC      CNT
          LAC      BIT
          ISZ      #CNT
          JMP      .-1
          RAL
          DAC      BIT
          LAS
          SMA
          JMP      LOOP
          JMP      GO
          START   GO
```

Características

- Reescrito posteriormente em linguagem C, tornou-se por essa razão **portátil**
- O novo código incorporava ao sistema o recurso da **multiprogramação**
- Permitia ainda **compartilhar procedimentos reentrantes**
- Esta nova versão do UNIX resultou por isso uns **30% maior** que a original

Distribuição

- UNIX foi então distribuído para dezenas de entidades sem fins lucrativos, em **1973**.
- A **versão 5** foi a primeira disponibilizada ao **público**, em 1975, e ainda permanece em uso.
- Em 1979, a **versão 7** se apresentou como uma versão **comercial** do produto.

Objetivos do projeto UNIX

- A primeira versão do UNIX foi totalmente desenvolvida por apenas **2 pessoas**
- Os maiores **usuários** eram os próprios **autores**
- Disso resultou uma forte **coerência**, presente no projeto do sistema assim obtido
- O principal objetivo foi manter o sistema **simples** e com **poucos recursos nativos**
- A **sofisticação**, externa ao sistema, foi deixada por conta de programas **aplicativos opcionais**

Generalidade

- O segundo objetivo foi a **generalidade**.
- Um mesmo procedimento reutilizado em diversas aplicações.
- As mesmas **chamadas de sistema** para arquivos, dispositivos, e comunicação entre processos;
- Sistemas de nomes, *aliasing*, proteções de acesso usados em arquivos, diretórios e dispositivos;
- Um **único mecanismo** para tratar **interrupções**, tanto de software como de hardware.

Reutilização e Composição

- O terceiro objetivo foi o de permitir a criação de **grandes programas combinando outros menores, já existentes.**

Acessos

- Apresenta estruturalmente três formas de acesso distintas:
 - Um *shell* (para comunicação com o operador),
 - Uma **biblioteca de rotinas de sistema**, e
 - Um conjunto de **chamadas de supervisor**

Processos

- O **núcleo** do UNIX representa **5 a 10% do sistema**: 10.000 linhas em C e mais 1.000 em linguagem de montagem.
- A **imagem do pseudo computador** do usuário representa um processo, e contém: imagem da memória, valores dos registradores, status dos arquivos abertos, diretório em uso, etc.
- As **imagens dos processos** permanecem residentes na memória durante a sua execução.

Mapa da Memória

- O mapa da memória do processo é dividido em **quatro segmentos**:
 - Segmento de **procedimentos reentrantes**,
 - Segmento de **dados**,
 - Segmento de **pilha**, e
 - **Área de crescimento** dos dois últimos.

Tabela de Texto

- Uma **tabela de texto** (*text table*) centraliza o controle das partes não modificáveis do código.
- Quando um processo solicita a execução de um **segmento compartilhado**:
 - Este é trazido do disco se não estiver presente na memória.
 - A **tabela de texto** recebe então um novo elemento, representativo desse segmento.

Segmento compartilhado

- Um **contador** associado ao segmento compartilhado indica o número de processos que estão usando tal segmento em cada instante.
- Quando este contador for **zerado**, o segmento **liberará a área de memória** que ocupava.

Segmento de Dados

- O **segmento de dados** compõe-se de uma parte relativa ao **sistema** e outra, relativa ao **programa**.
- Na parte do sistema, ficam registradas:
 - Informações sobre o conteúdo dos registradores,
 - Os descritores de arquivos,
 - Dados de contabilidade,
 - Rascunhos, e
 - Uma pilha, para uso do sistema

Tabela do Processo

- Cada processo tem uma **tabela do processo**, acessada pelo núcleo do sistema, preservando todos os dados necessários acerca do processo quando ele não está ativo:
 - **Nome** do processo,
 - Localização dos seus **segmentos**,
 - Informações de ***scheduling***.

Operações

- **Operações do gerenciamento de processos:**
 - Criar processo,
 - Executar processo,
 - Sincronizar processos,
 - Escalar processo,
 - *Swapping*,
 - Terminação do processo.

Entrada e Saída

- Dois componentes:
 - **block** (*structured*) I/O e
 - **character** (*non-structured*) I/O.
- **System calls:**
 - **gtty, stty, stat:**
(**get, set, read** dados sobre arquivos e terminais).
 - **abrir/fechar arquivo:**
 - **fd=open**(filename, mode) [fd = file descriptor]
 - **close**(fd)
 - **read/write:** nbyteslidos = **read** (fd,buffer,nbytesdesejados)
(a leitura ou gravação é **sequencial**).
 - **posicionar** para i/o: **seek** (fd, posição, tipo) onde tipo:
relativo/absoluto, número de bytes por transferência.

Sistema de Arquivos

- Os arquivos podem conter **qualquer tipo de dados** que o usuário neles quiser depositar.
- Os registros dos arquivos **não precisam ter nenhuma estrutura imposta** pelo sistema.
- O Sistema de arquivos é residente em disco; os **blocos têm 512 bytes**.
- **Diretório hierárquico**
(caminho: x/y/z - raiz x, sub-raiz y , folha z).

Disco

- O disco é dividido em 4 regiões:
 - **Área de boot;**
 - Área contendo o tamanho do disco e os limites das demais regiões;
 - **Área do *i-list***, com a lista dos ***i-nodes***, cada qual descrevendo um arquivo.
 - Cada i-node é o descritor de um arquivo, e contém todas as informações sobre ele, exceto seu nome e os dados;
 - **Área para os arquivos.**

- ***I-node*** contém:
 - Identificação do **proprietário** do arquivo;
 - Identificação do **projeto**;
 - Bits de **proteção**;
 - **Endereços** dos conteúdos do arquivo;
 - **Tamanho** do arquivo;
 - Instante da **criação**;
 - Instante do **último acesso**;
 - Instante da **última alteração**;
 - **Número de links** para esse arquivo;
 - **Tipo** de arquivo: diretório, comum, especial.

Shell

- Proporciona ao usuário uma **interface de controle**.
- Cada comando tem um **nome** e uma **lista de argumentos** específicos, separados por brancos.
- Arquivos precedidos de < fornecem **entradas** para um comando.
- Arquivos precedidos de > recebem as **saídas** do comando.
- Arquivos de saída precedidos de >> **acrescentam** dados (*append*) em vez de sobrescrever.

- Coringas
 - Asterisco * aceita **qualquer cadeia** de caracteres
 - Interrogação ? denota **qualquer caractere**
- **Pipes** (definem linhas de produção)
 - São arquivos abertos conectando dois processos: **a | b**
 - As saídas de cada programa se tornam as entradas do programa seguinte

- Sincronização, *scheduling*, *buffering* são feitos automaticamente pelo sistema
- **Pipeline** é uma fila linear de *pipes*: **a | b | c | d**
- Processos podem encadear-se de forma mais geral.
- **Filtros** processam um fluxo de entrada, produzindo um fluxo de saída

- ***Multitasking*** pode ser especificado em UNIX usando-se um “&” após o comando que se deseja executar concorrentemente. Por exemplo:
 - **(a;b;c)&** indica que a sequência “primeiro **a**, depois **b**, finalmente **c**” vai ser executada em *background*, concorrentemente com outras que porventura tenham sido previamente ativadas.

unix

Esta apresentação sobre UNIX pode ser complementada com material de

Andrew S. Tanenbaum

Sistemas Operacionais Modernos

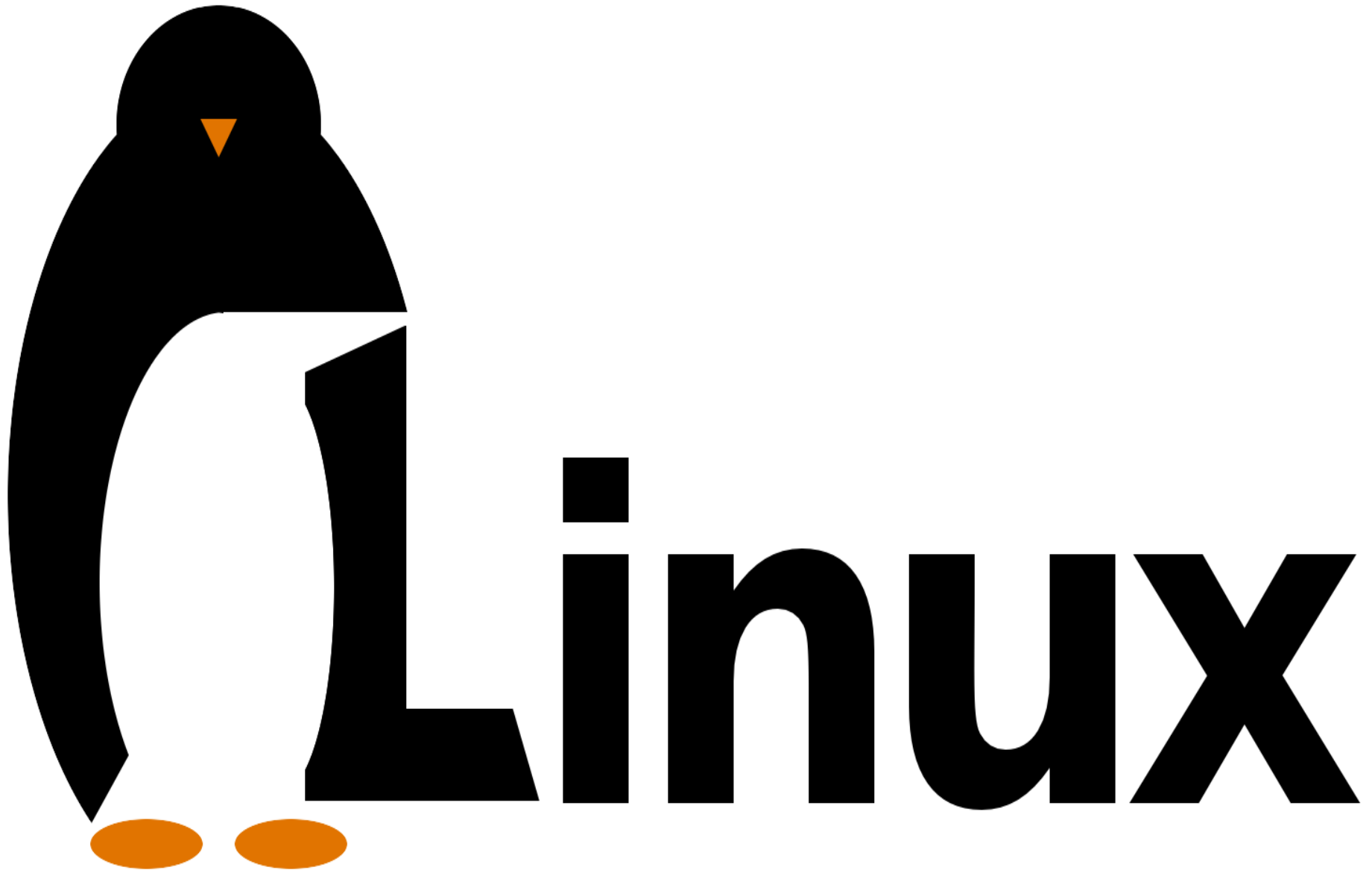
Pearson Education do Brasil, 3ª. Edição, 2009.



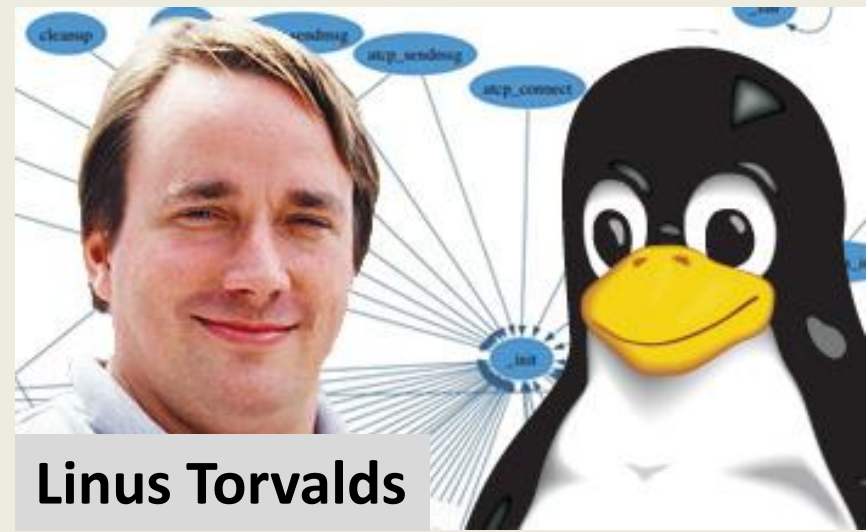
Material complementar, referente ao sistema Linux versão 2.0, de 1996
Este material não foi apresentado em classe.

COMPLEMENTO 1

O SISTEMA OPERACIONAL LINUX



Histórico



Linus Torvalds

- **LINUX** é um dos vários Sistemas Operacionais desenvolvidos para serem compatíveis com o UNIX
- Este sistema tem **evoluído** continuamente, e de forma muito rápida.
- Este resumo refere-se à **versão 2.0**, de 1996
- Seu desenvolvimento começou em **1991**
- **Linus Torvalds**, finlandês, desenvolveu o primeiro Linux para o processador **80386 da Intel**, de 32 bits.

O processador Intel 80386



Núcleo (*kernel*)

- Com o apoio de uma comunidade da Internet, o Linux cresceu a partir de um *kernel* inicial, até realizar boa parte dos serviços do Unix.
- Inicialmente, o trabalho girou somente em torno do *kernel*, um módulo original do sistema.
- O Linux, como o conhecemos hoje, agregou desde então muitos módulos adicionais, alguns originais, outros adaptados.







Outras Características

- O ambiente Linux é **padronizado**, mas não padroniza o uso dos aplicativos.
- Uma **distribuição** Linux incorpora tipicamente o Linux básico, e um conjunto de ferramentas de **instalação e atualização**.
- Para obter melhor **desempenho**, inicialmente o sistema foi implementado **monoliticamente**.

Projeto modular

- Permite carregamento e descarregamento **dinâmico** de *drivers*, em **tempo de execução**.
- É um sistema **multiusuário**, com **proteção** entre processos.
- Usa um ***scheduler*** para partilhar o tempo do processador entre os processos ativos.
- Novos processos **compartilham** seletivamente partes do seu ambiente ancestral.
- Isso permite trabalhar em ***multi-threading***.

Protocolos e Comunicação

- Os processos se comunicam através de um **sistema de comunicação** herdado do **System V** (usando filas de mensagens, semáforos, memória compartilhada) e pela interface do BSD Linux (*Berkeley Software Distributions*)
- **Vários protocolos** de rede estão **simultaneamente** disponíveis no sistema.

Sistemas de Arquivos

- O sistema de arquivos é **hierárquico** para o usuário, como no Unix.
- Internamente, outra camada torna possível o **gerenciamento simultâneo** de diferentes sistemas de arquivos.
- Há também no Linux recursos que dão suporte para **sistemas de arquivos virtuais**.

Caches de Páginas e de Buffers

- Sistemas de arquivos baseados em dispositivos acessam o disco através de **dois *caches***:
 - os dados usam o *cache de páginas*, em comum com a memória virtual, e
 - os meta-dados usam um *cache de buffers* separado, indexado pelo respectivo bloco físico do disco.

Gerenciamento de Memória (virtual)

- O gerenciamento de memória **compartilha páginas** e usa mecanismo ***copy on write*** para minimizar a duplicação de dados compartilhados.
- Essas páginas são carregadas segundo a necessidade formando um sistema de **paginação** no disco.
- O método de substituição de páginas utilizado é o **LFU – *least frequently used***: escolhe para serem substituídas em primeiro lugar páginas presentes na memória que tenham sido usadas mais raramente.

Aspectos do Projeto

- É essencialmente um **Unix sem o seu *microkernel***
- ***Multiusuário e Multitarefa***, compatível com Unix
- Disponível para **muitas plataformas**
- Executado **eficientemente** em máquinas com **muita memória e disco**, mas consegue operar em sistemas relativamente pequenos, a partir de 4MB de RAM
- Os ***Kernels*** mais **antigos** eram **minimalistas**, mas foram crescendo com o barateamento do hardware

Linux kernel diagram

functions

system

networking

storage

memory

processing

human
interface

layers

user space
interfaces

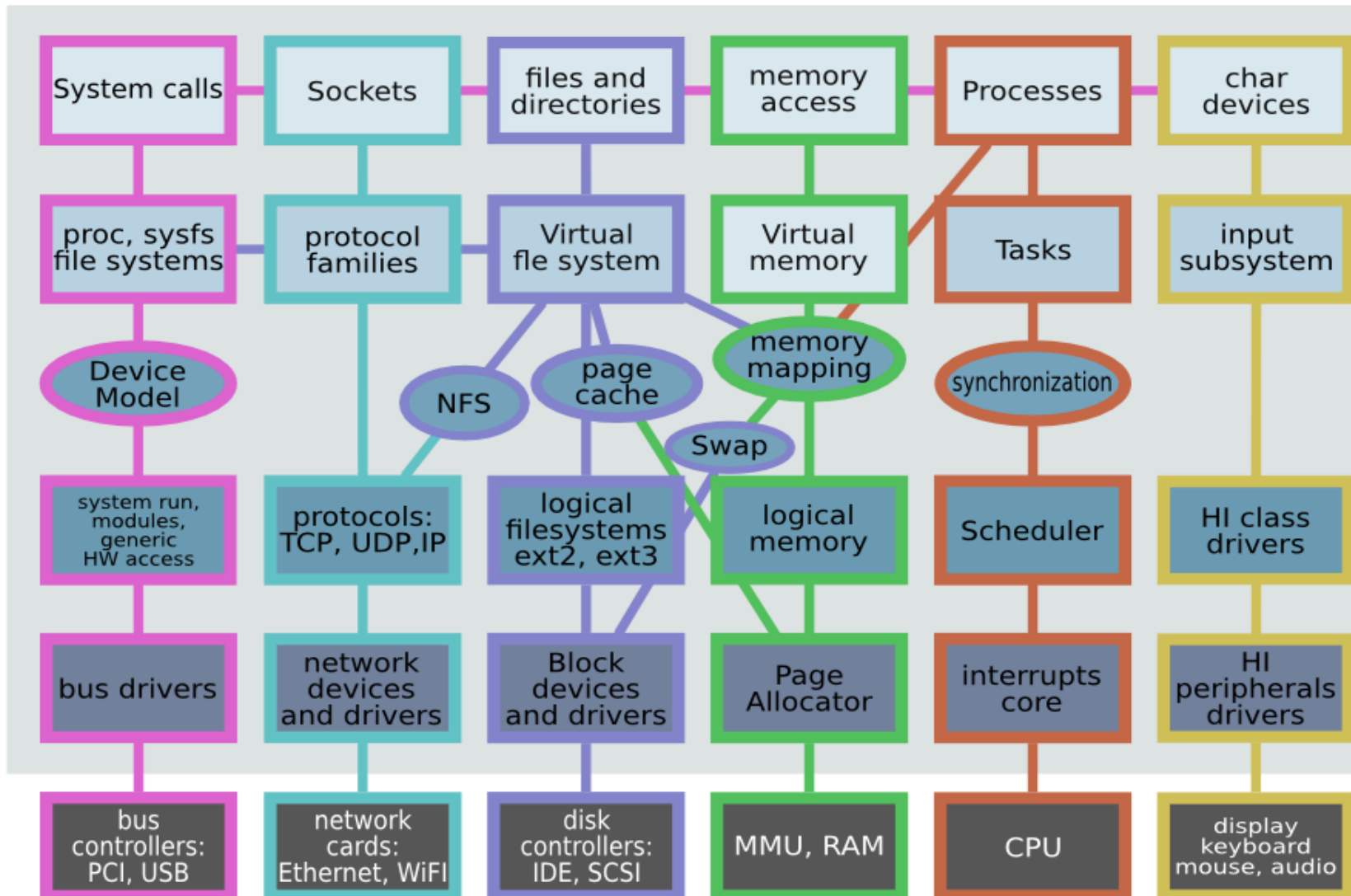
virtual
subsystems

bridges

logical

hardware
interfaces

electronics



© 2007-2009 Constantine Shulyupin <http://www.MakeLinux.net/kernel/diagram>

Linux™



**Esta apresentação sobre o sistema
Linux baseia-se no material de:**

**Silberschatz, Galvin e Gagne
Applied Operating Systems Concepts
Wiley, 2000**





Este material não foi apresentado em classe.

COMPLEMENTO 2

SISTEMA OPERACIONAL GNU

What is GNU?

- GNU is an operating system that is [free software](#)—that is, it respects users' freedom.
- The development of GNU made it possible to use a computer without software that would trample your freedom.
- We recommend [installable versions of GNU](#) (more precisely, GNU/Linux distributions) which are entirely free software.

GNU



- GNU is a Unix-like operating system.
- That means it is a collection of many programs: applications, libraries, developer tools, even games.
- The development of GNU, started in January 1984, is known as the GNU Project.
- Many of the programs in GNU are released under the auspices of the GNU Project; those we call [GNU packages](#).



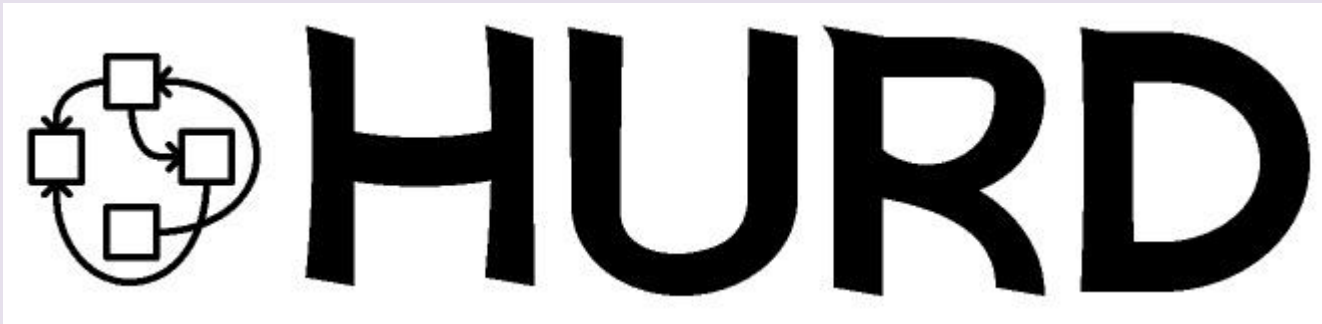
- The name “GNU” is a recursive acronym for “GNU's Not Unix.”
- “GNU” is pronounced *g'noo*, as one syllable, like saying “grew” but replacing the *r* with *n*.
- The program in a Unix-like system that allocates machine resources and talks to the hardware is called the “kernel”.



GNU/Linux



- GNU is typically used with a kernel called Linux.
- This combination is the GNU/Linux operating system.
- GNU/Linux is used by millions, though many call it “Linux” by mistake.



- GNU's own kernel, [The Hurd](#), was started in 1990 (before Linux was started).
- Volunteers continue developing the Hurd because it is an interesting technical project.



GNU SOFTWARE

- GNU is an operating system which is 100% free software.
- It was launched in 1983 by Richard Stallman
- It has been developed by many people working together for the sake of freedom of all software users to control their computing.
- Technically, GNU is generally like Unix.
- But unlike Unix, GNU gives its users freedom.

- The GNU system contains all of the [official GNU software packages](#)

<https://www.gnu.org/philosophy/categories.html>

- It also includes non-GNU free software, notably TeX and the X Window System.
- Also, the GNU system is not a single static set of programs.
- Users and distributors may select different packages according to their needs and desires.
- The result is still a variant of the GNU system.

- If you're looking for a whole system to install, see our [list of GNU/Linux distributions which are entirely free software](https://www.gnu.org/distros/free-distros.html):

<https://www.gnu.org/distros/free-distros.html>

- To look for individual free software packages, both GNU and non-GNU, please see the [Free Software Directory](http://directory.fsf.org/): a categorized, searchable database of free software:

<http://directory.fsf.org/>

- The Directory is actively maintained by the [Free Software Foundation](http://www.fsf.org/) and includes links to program home pages where available, as well as entries for [all GNU packages](http://directory.fsf.org/wiki/GNU/):

<http://directory.fsf.org/wiki/GNU/>

- Another list of [all GNU packages](https://www.gnu.org/software/software.html) is below:

<https://www.gnu.org/software/software.html>

- [Free software documentation links](https://www.gnu.org/doc/doc.html) are listed separately:

<https://www.gnu.org/doc/doc.html>

- Finally, we have a [short list of free software for Microsoft Windows](https://www.gnu.org/software/for-windows.html), for Windows users who would like to try free software:

<https://www.gnu.org/software/for-windows.html>