



Universidade do Estado do Rio de Janeiro  
Faculdade de Engenharia  
Laboratório de Engenharia Elétrica

# Introdução ao uso do Linux (parte 1)

Autores: [Elaine de Mattos Silva](#)

Raphael Guedes

Orientador: Prof. José Paulo Vilela Soares da Cunha

Rio de Janeiro, 23 de setembro de 2005

Programa de Estágio Interno Complementar do CETREINA/SR-1/UERJ

# Contatos

- Apostila e apresentações disponíveis em:  
<http://www.lee.eng.uerj.br/~elaine>
- e-mail de contato:  
[elaine.mattos@gmail.com](mailto:elaine.mattos@gmail.com)  
[elaine@lee.eng.uerj.br](mailto:elaine@lee.eng.uerj.br)

# Objetivos do curso

- Motivar o contato com Linux e outros *Softwares Livres*
- Difundir o uso do Linux entre os alunos de Engenharia Elétrica
- Formar multiplicadores da cultura de *software* livre

# Organização deste curso

- Aula 1
  - *Software Livre*
  - Evolução dos sistemas operacionais
  - Noções básicas
- Aula 2
  - Principais comandos
- Aula 3
  - Principais comandos (cont.)
  - Programas de interface gráfica

# Aula 1

- Aula 1
  - *Software* Livre/Linux
    - Por que usar
    - Algumas aplicações
  - Evolução dos sistemas operacionais
    - Um sistema computacional
    - Sistemas Operacionais
    - UNIX/Linux
    - *Software* Livre

# Aula 1 (cont.)

- Aula 1
  - Noções básicas
    - *Shell*
    - *Kernel*
    - Distribuições
    - Interfaces Gráficas
    - Sistemas de Arquivo
    - Estrutura de Diretórios
    - Modos de Permissão

# *Software* Livre – Por que usar?

- Acesso ao código fonte
- Qualidade / baixo custo
- Livre de pirataria

# Algumas aplicações: Planilha feita com *OpenOffice*

Tratamento de dados 1.005 - Correção: Office 3.1.1.2

Arquivo Editar Exibir Inserir Formatar Ferramentas Dados Janela Ajuda

/cariora/br/nevernoppix/cursos/linux/figs/total

Arial

4

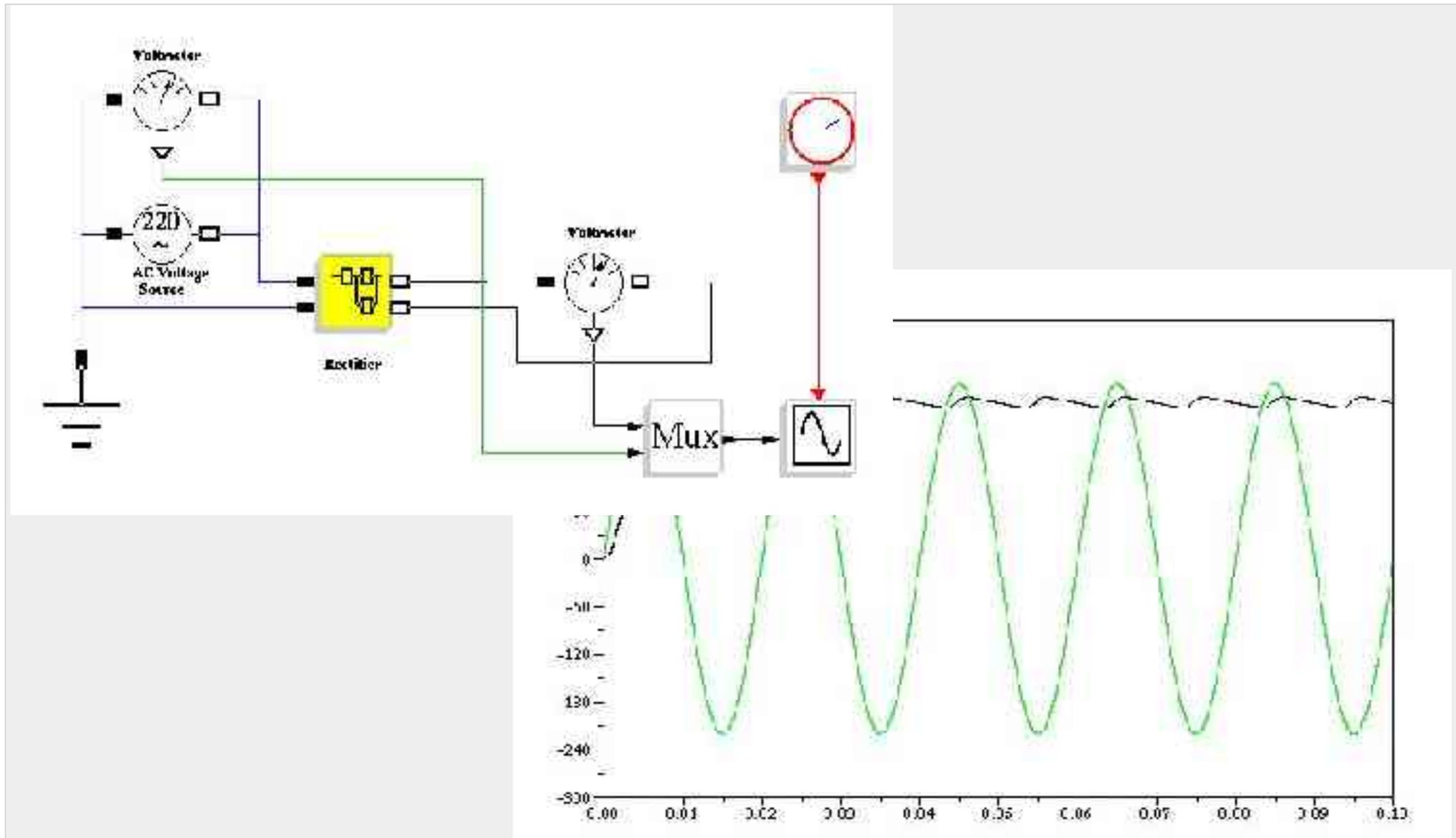
Método Dos Mínimos Quadrados (Transformadores)											
n° de medições	V	X	X <sup>2</sup>	XY	Y <sup>2</sup>	V/(mx-b)	m experimental	Mxx	Mxy	MMQ	
1	0,00	7,00	4,00	0,00	4,00	0,40	0,3	49,00	0,00	4,00	
2		15,00	2,00	23,00	225,00	0,72				15,00	
3		23,00	6,00	38,00	529,00	0,00	b			23,00	
4		31,00	8,00	68,00	961,00	0,01	1,15			31,00	
5		39,50	10,00	108,16	1560,25	0,00				39,50	
6		47,80	12,00	156,25	2284,64	0,00	m esperado			47,74	
7		55,80	14,00	213,16	3113,64	0,02	4,00			55,95	
8		64,00	16,00	278,89	4096,00	0,03				64,17	
9		72,50	18,00	352,44	5256,25	0,01	b esperado			72,33	
10		80,80	20,00	436,81	6528,64	0,04	0,30			80,59	
<b>Resumo</b>											
11											
12		432,10	114,00	200	6425,21	200	0,64				
13											
14											
15											
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											
27											
28											
29											
30											
31											
32											
33											
34											
35											

Desvio Padrão			
erro m	erro b		
2,0880	0,0157	0,2029	

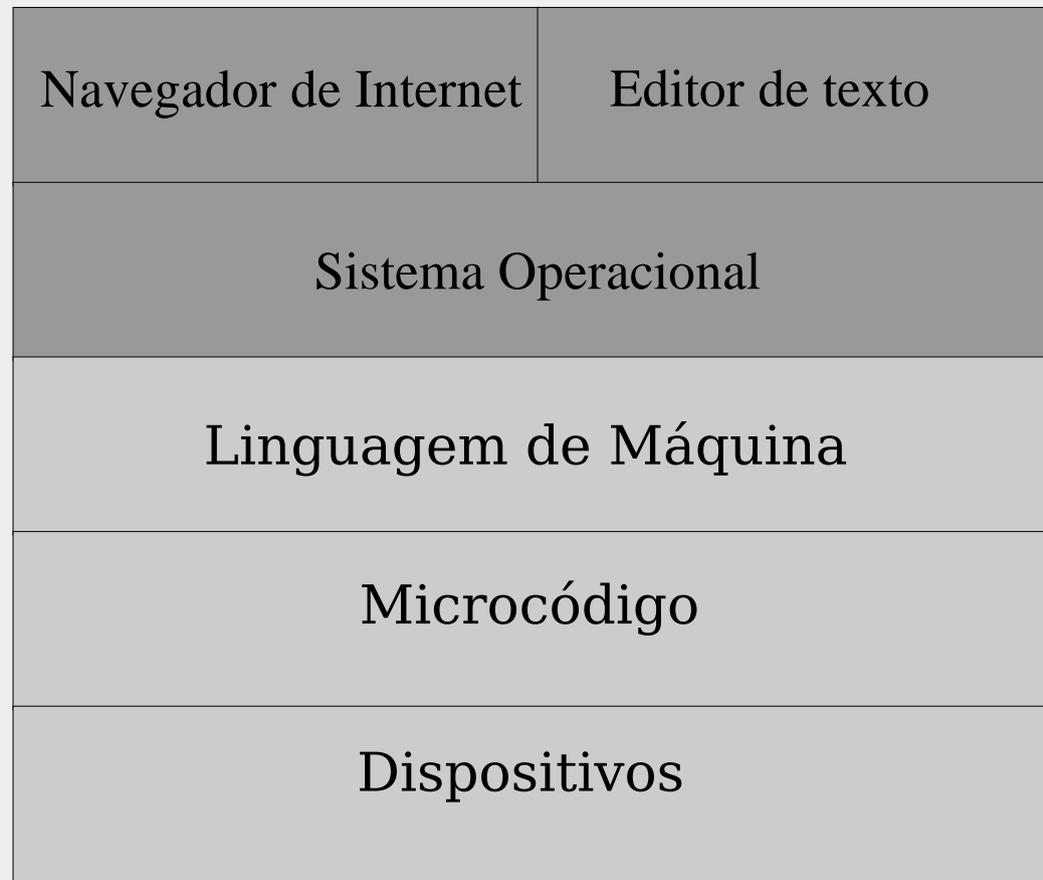
Comparação dos erros			
m			
###	é maior que	###	OK
###	é menor que	1,9770	ERRO
b			
0,00	é maior que	-0,35	ERRO
0,00	é menor que	-0,35	ERRO

Planilha 1.01 | As tabelas | 100% | DESV | soma: 0,00

# Algumas aplicações : Simulação de circuito com *Scilab*



# Um Sistema Computacional



 *Software*

 *Hardware*

# Evolução dos Sistemas Operacionais

- Na década de 60
  - *software* que controla o *hardware*
- Atualmente
  - camada de abstração de *hardware*
  - gerenciador de recursos

# Evolução dos Sistemas Operacionais<sub>(cont.)</sub>

- Tarefas Principais
  - interface para o usuário
  - gerencia recursos de *hardware* entre programas
  - permite compartilhamento de dados entre usuários
  - facilita tarefas paralelas, etc

# Evolução dos Sistemas Operacionais<sub>(cont.)</sub>

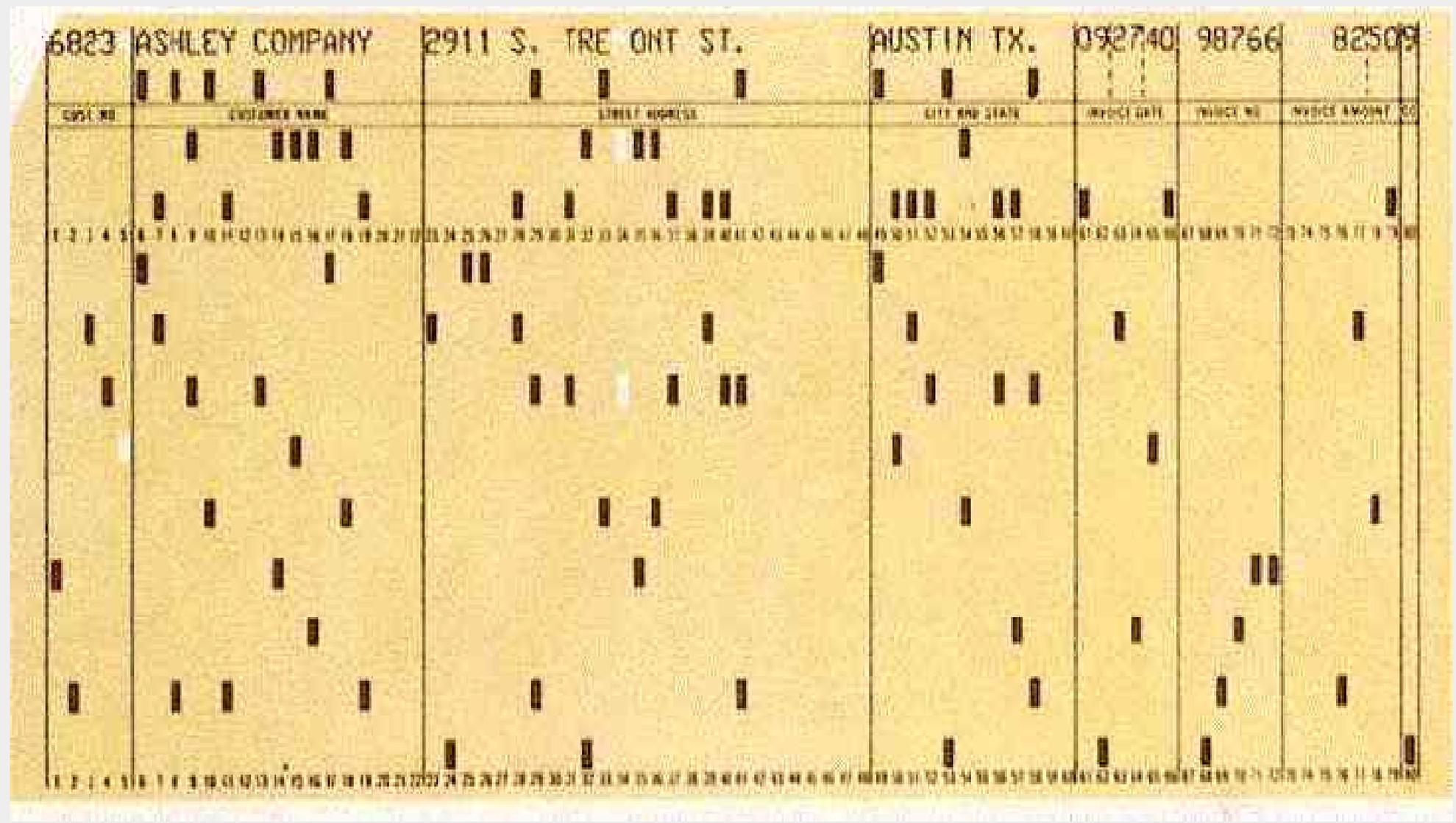
- **Década de 40**
  - máquinas de cálculo numérico
  - 20.000 válvulas e relés
  - programação através de conexões elétricas

# Evolução dos Sistemas Operacionais<sub>(cont.)</sub>

## ▪ Década de 50

- transistores substituem válvulas
- programação em *assembly* ou ForTran (formula translator)
- cartões perfurados
- computador central
- processamento em lotes

# Um cartão perfurado



# Evolução dos Sistemas Operacionais<sub>(cont.)</sub>

- **Década de 60**

- família *System/360* da IBM
- OS/360 – escrito em *assembly*
- novos conceitos em programação
- processamento em lotes

# Evolução dos Sistemas Operacionais<sub>(cont.)</sub>

## ▪ Década de 60

- M.I.T. desenvolve sistema CTSS
- conceitos: terminal e processamento *background*
- M.I.T./Bell Labs/G.E. criam MULTICS
- linguagem: PL/I da IBM

# Evolução dos Sistemas Operacionais<sub>(cont.)</sub>

- **Década de 60/70**

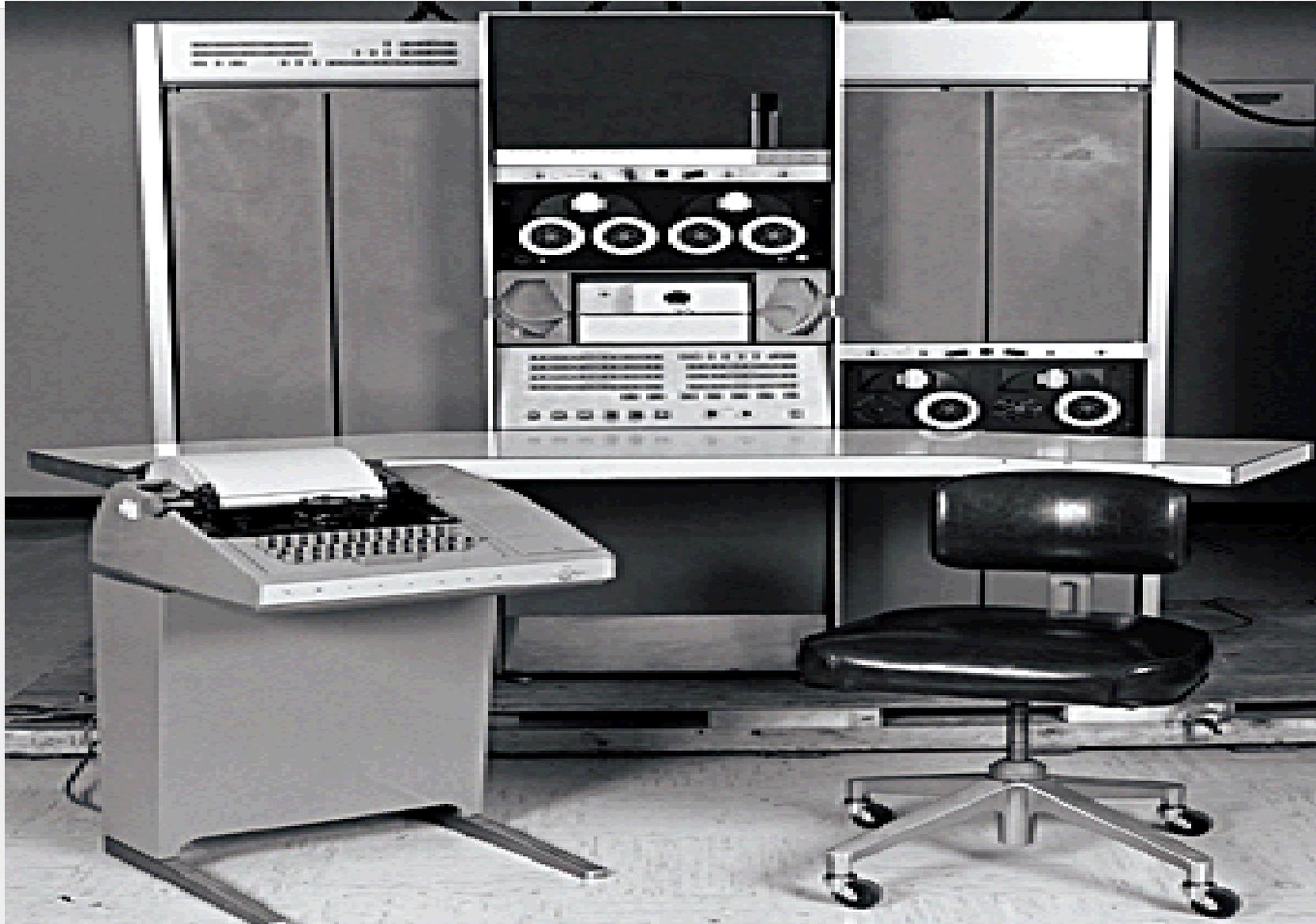
- Ken Thompson (*Bell Labs*)

- MULTICS em *assembly* para DEC PDP-7

- Brian Kernighan (*Bell Labs*)

- UNICS = *UNIX*

# PDP-7



# Ken Thompson e Brian Kernighan



# Evolução dos Sistemas Operacionais<sub>(cont.)</sub>

- **Década de 60/70**
  - UNIX não era portátil
  - Dennis Ritchie (*Bell Labs*)
    - aperfeiçoa linguagem B e chama de C
  - Ritchie e Thompson
    - reescrevem UNIX em C

# Evolução dos Sistemas Operacionais<sub>(cont.)</sub>

- **Década de 70/80**
  - *Bell Labs* subsidiária AT&T
    - não podia *vender* licença do Unix
  - com código-fonte aberto surgiram
    - Berkeley Software Distribution - BSD
    - XENIX– *Microsoft*
    - AIX– *IBM*
    - HP-UX - *HP*

# Evolução dos Sistemas Operacionais<sub>(cont.)</sub>

- **Década de 70/80**
  - quebra da AT&T em pequenos blocos
  - fechado acesso ao código-fonte
  - IEEE cria padrão POSIX

# Evolução dos Sistemas Operacionais<sub>(cont.)</sub>

- **Década de 80/90**
  - Andrew Tanenbaum
    - MINIX: mini UNIX
    - escrito em C
    - código legível (feito para estudo de S.O.'s)
    - sem aplicação comercial
    - não é cópia do UNIX

# Evolução dos Sistemas Operacionais<sub>(cont.)</sub>

- **Década de 90**
  - Linus Torvalds
    - **LINUX: clone MINIX**
      - mais recursos
      - aplicações comerciais

# *Software* Livre

- 1984 – Richard Stallman / M.I.T
- caso da impressora
- versões de código aberto para vários programas
- criação da *Free Software Foundation-FSF*

# *Software Livre* (cont.)

- Criação de vários *softwares* para sistemas *UNIX-Like*
- *GNU\*-Public License (GPL)*
- filosofia do *software livre*

\*GNU – Gnu is Not Unix

# Noções Básicas (cont.)

- *case sensitive*
- extensões de arquivos flexíveis e não obrigatórias

# Noções Básicas (cont.)

- sistema multiusuário
- conta de usuários
- super usuário = *root*

# Noções Básicas (cont.)

- Permissão de arquivos e diretórios
  - **r** – (*read*): leitura
  - **w** – (*write*) : escrita
  - **x** – (*execute*) : execução

# Noções Básicas (cont.)

- Acesso direto aos dispositivos (diretórios)

Ex.:

- cd-rom: `/dev/cdrom`
- disco rígido primário: `/dev/hda`

# Noções Básicas : Distribuições

- $\text{Linux} = \textit{kernel}$
- $\textit{kernel} + \text{aplicativos} = \text{distribuição}$

# Noções Básicas : Distribuições (cont.)

- Principais Distribuições:
  - Debian
    - Debian-BR-CDD - versão brasileira voltada para usuários finais
  - Red Hat
    - versão popular e base de outras distribuições

# Noções Básicas : Distribuições (cont.)

- Mandrake
  - distribuição da empresa francesa Mandrakesoft
  - fácil uso e bom número de aplicativos
- Conectiva
  - versão nacional originada da Red Hat voltada para o mercado nacional
  - recentemente a Mandrakesoft e a Conectiva se fundiram e criaram a Mandriva.

# Noções Básicas : Distribuições (cont.)

- Kurumin
  - versão nacional muito amigável baseada em Debian
- Karioca
  - projeto de graduação de aluno da UERJ
- Slackware
  - uma das primeiras distribuições (1993)
  - similar ao UNIX, muito personalizável porém com poucas ferramentas automáticas

# Noções Básicas : *Kernel*

- parte central do S.O.
- gerencia recursos da máquina (processos, memória, E/S)
- pode ser considerado o Linux em si

# Noções Básicas : Versões do *Kernel*

- Primeiro *kernel* (Torvalds) versão 0.02
- Tipo de numeração atual
  - a. b. c – onde
    - a- série do *kernel*.
    - b- série do *kernel* – se ímpar, em desenvolvimento
    - c - versão daquela série

## Noções Básicas : Versões do *Kernel* (cont.)

Ex.:

```
[aluno@localhost aluno]$ uname -r  
2.6.8.1-12mdk  
[aluno@localhost aluno]$
```

- série 2.6, versão 8

# Noções Básicas : *Shell*

- *shell* – interpretador de comandos
- inicializado após o *login*
- interface *kernel*/usuário
- acesso via terminal ou gráfico

# Noções Básicas : *Shell*

## Principais *Shells*

- *Bourne Shell*
  - desenvolvido por Stephen Bourne
  - UNIX
  - /bin/sh
- *Bourne-Again Shell*
  - Linux
  - *Bourne Shell* melhorado
  - /bin/bash

# Noções Básicas : *Shell* (cont.)

- *Korn Shell*
  - *upgrade* do *Bourne Shell*
  - desenvolvido por David Korn
  - `/bin/ksh`
- *C Shell*
  - desenvolvido por Bill Joy
  - estrutura similar ao C
  - `/bin/csh`

# Noções Básicas : Visão Geral do Sistema



# Noções Básicas : *Shell Script*

- arquivo texto com comandos ou instruções que serão executados pelo *shell*
- facilitam tarefas de administração do sistema, tarefas repetitivas e outros

# Noções Básicas : *Shell Script*

## Exemplo de *script*

```
#!/bin/bash  
clear  
echo "Versao do Kernel: "; uname -r  
echo "O arquivo do kernel esta em : ";  
find /boot/vmlinuz-  
exit 0
```

# Noções Básicas : Interfaces Gráficas

- sistema de janelas – M.I.T. e DEC projeto *X Window System* ( UNIX mais acessível)
- surgiram diversos gerenciadores de janelas

# Noções Básicas : Interfaces Gráficas (cont.)

- **Gerenciadores de Janelas**
  - controlam interação com usuário
- **Principais gerenciadores**
  - GNOME
  - interface intuitiva do projeto GNU
  - *K Desktop Environment* -KDE
  - opção mais popular

# Noções Básicas : Sistemas de Arquivos

- estrutura de armazenamento e manipulação de dados
- todo S.O. possui sistemas de arquivos
- sistemas de arquivos *UNIX-Like* são hierárquicos

# Noções Básicas : Sistemas de Arquivos ((cont.))

- **Principais sistemas de arquivos**
  - *MS-Windows*
    - FAT – *file allocation table*
    - NTFS – *network file system*
  - *UNIX- Like*
    - ext3
    - ReiserFS
    - MINIX

# Noções Básicas : Sistemas de Arquivos ((cont.))

- minix – nativo do MINIX, primeiro sistema do Linux. Substituído em abril de 1992
- ext – *extended file system*. Upgrade do minix com suporte a discos maiores e maior performance
- ext2 – segunda versão do ext. Dá suporte a arquivos de até *2 terabytes* e nomes de arquivos com 255 caracteres

# Noções Básicas : Sistemas de Arquivos ((cont.))

- ext3 – terceira versão do ext. Inclui sistema *journaling*
- Reiserfs – criado por Hans Reiser. Tamanho de arquivo 8TB e HD de 16TB também implementa *journaling*
- Sistema de arquivos *journaling* – gera um arquivo de log , o journal, onde salva alterações, para depois gravá-las no sistema de arquivo

# Noções Básicas : Sistemas de Arquivos ((cont.))

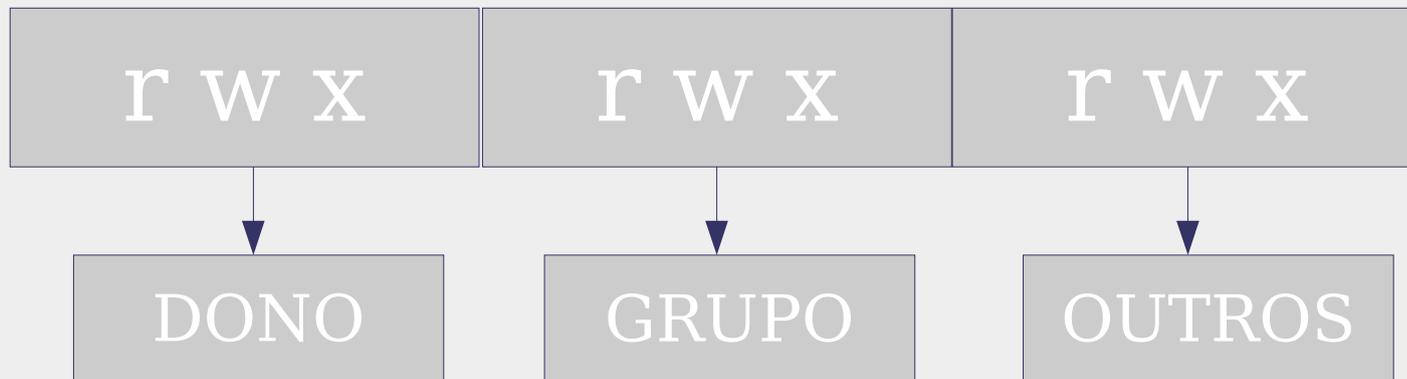
- *inode* ( ou *i-node*) – estrutura de dados que contém informações sobre um arquivo ou diretório
  - UID e GID
  - Tipo e permissões do arquivo
  - Data e hora de criação
  - Tamanho do arquivo
  - Localização do arquivo

# Noções Básicas : Estrutura de Diretórios

- Estrutura hierárquica com funções definidas
  - / - raiz
  - /home - diretórios dos usuários ex.: [/home/elaine](#)
  - /root - home do super usuário
  - /usr - programas dos usuários
  - /bin - executáveis e comandos mais comuns
  - /boot - arquivos do *kernel*
  - /etc - arquivos de configuração e administração
  - /mnt - montagem temporária ex.: [cd rom, floppy, etc](#)

# Noções Básicas : Modos de Permissão

- No Linux arquivos e diretórios são protegidos
- Três grupos: dono, grupo, outros
- Três permissões: leitura (r), escrita (w), execução (x)



# Na aula 2 veremos:

- ajuda e informações
- navegação pelos diretórios
- montagem de sistemas de arquivos
- localização e manipulação de arquivos
- comandos de paginação
- compactação e descompactação
- gerenciamento de processos, grupos e privilégios
- comandos para impressão

# Referências

- Material didático
  - Apostila - Introdução ao Uso do Linux
    - <http://www.lee.eng.uerj.br/~elaine>
- Mais sobre *Software Livre*
  - Documentário *Revolution OS*
  - Cartilha de *Software Livre*
- Evolução dos Sistemas Operacionais
  - Livro: A. S. Tanenbaum. *Sistemas Operacionais Modernos*. Prentice Hall do Brasil, Rio de Janeiro, 1995

# Referências (cont.)

- *Shell Script*

- Livro: J.C. Neves. *Linux: Programando em Shell*.  
Brasport, Rio de Janeiro, 2000.

- Tutorial: Programação em *Bourne-Again Shell*

- <http://meleu.da.ru>



- <http://www.openoffice.org>



- <http://www.scilab.org>