

Aula 1

Administração de redes Linux
Laboratório Daniel Bauman Henry
LabMAP



MAP - IME - USP

Agenda

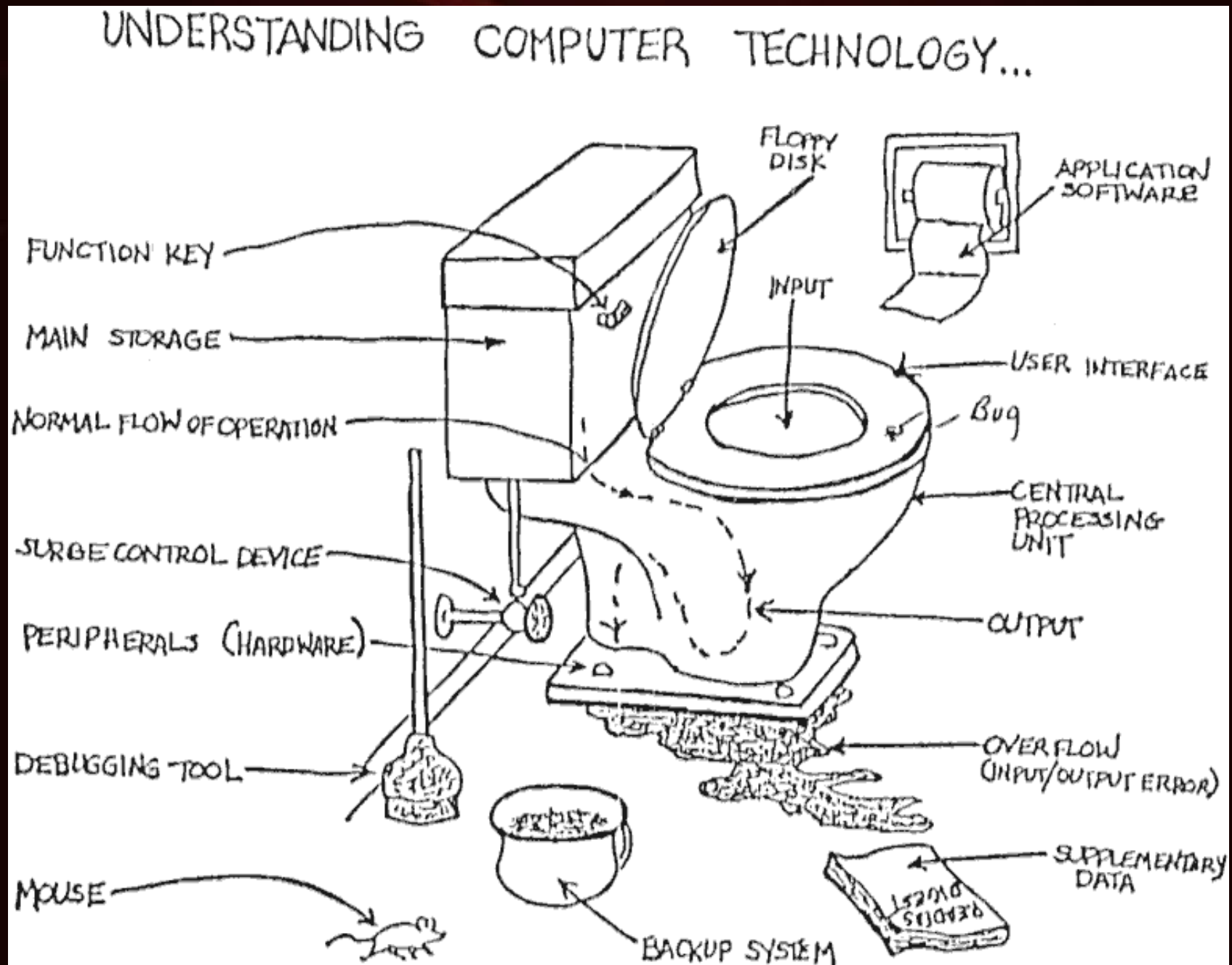
1. Apresentação do curso
2. História e filosofia do Linux
3. Breve introdução ao Slackware
4. Alguns conceitos básicos
5. Bash
 - Comandos básicos (cd, ls, grep, tar...)
 - Caracteres coringa
 - Editores de texto
 - Criação de scripts
 - Pipe e redirecionamento de entrada/saída
 - Gerenciando processos
 - Logs
6. Lista de exercícios

Apresentação

Então você quer administrar uma rede Linux?



Apresentação



Linux e GNU/Linux

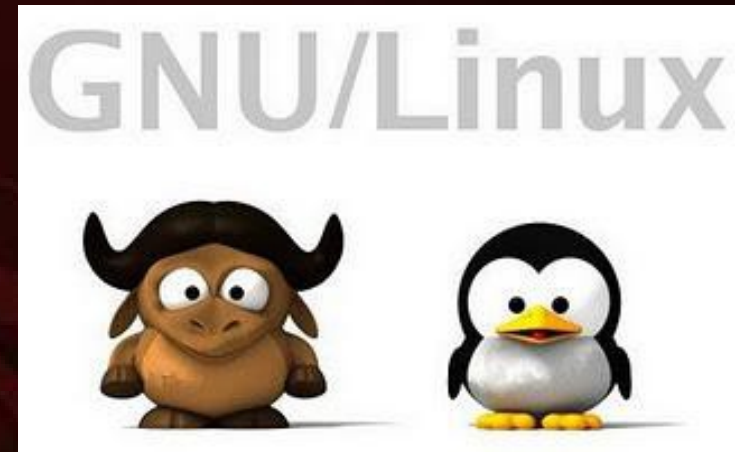
```
if(linux == kernel)
    printf("You got a snack!");
```

Kernel

- Interface com o hardware
- Projeto iniciado em 1991 por Linus Torvalds que era um tremendo mão de vaca
- Sozinho o kernel não tem muita utilidade

GNU/Linux

- Kernel + gcc + bash + K⁴



Slackware Linux

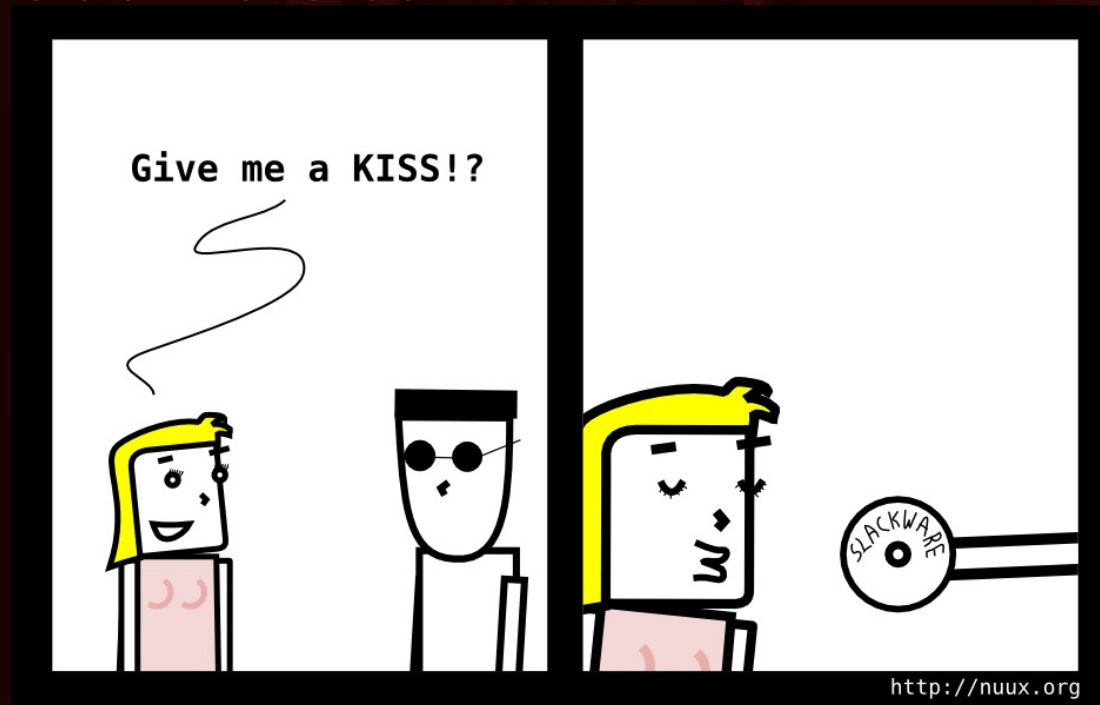


slackware
linux



Slackware Linux

- Lançado em 1993, maior de idade
- É a distro mais antiga ainda mantida
- Só Slackware é Linux e Volkerding é seu profeta
- Distro chart
- <http://www.slackbook.org>
- <http://www.slackware.com>



Bash



- Shell: interface com SO interpretando comandos
- Bash: um shell GNU, baseada na Bourne Shell, amplamente usada

Root

- Acesso local irrestrito
- Não diga o nome do root em vão
- Poderes supra-divinos
- ID 0



Linha de comando

```
oda@thor:~$ cat DONT_FEAR_THE_COMMAND_LINE
```

[illegible]

```
oda@thor:~$ AAAAAaaaaaa.....
```

Alguns comandos básicos

- man

- Exibe o manual de algum comando, programa ou biblioteca
- Modo de uso:
 - `man [nº da seção] <cmd>`
- Dividido em sete seções (alguns itens possuem mais de uma entrada em seções diferentes, por ex, `printf`)
- Mais detalhes: `man man` (funciona!)

- apropos

- Faz uma busca nas man pages
- Modo de uso:
 - `apropos <key word>`

- info



Alguns comandos básicos

- cd
 - Muda o diretório (cd vem de change working directory).
 - Modo de uso:
 - cd <caminho do diretório>
 - Para saber o diretório atual, use o comando pwd



Alguns comandos básicos

- ls

- Lista os arquivos de um diretório
- Modo de uso:
 - ls [caminho para o diretório]
- Alguns parâmetros úteis:
 - ls -l: exibe mais detalhes sobre os arquivos
 - ls -a: lista também os arquivos ocultos
- Mais detalhes: man ls

- cat

- Concatena arquivos e exibe o resultado na saída padrão
 - Modo de uso:
 - cat [arquivo1] [arquivo2] ...
- Se não receber nenhum arquivo como parâmetro, lê da entrada padrão



Alguns comandos básicos



Alguns comandos básicos

- Antes de seguir, vamos falar sobre permissões:
 - Nos sistemas de arquivo do linux, cada arquivo possui três tipos de permissão:
 - leitura (r), escrita (w) e execução (x)
 - O controle dessas permissões acontece em três níveis:
 - usuário (u), grupo (g) e outros (o)
 - Portanto, para guardar essas informações, são necessários 9 bits
 - Cada grupo de três bits pode ser representado com um algarismo na base 8, portanto podemos representar as permissões de um arquivo com 3 algarismos octais



Alguns comandos básicos

Agora podemos seguir para o próximo comando:

- **chmod**

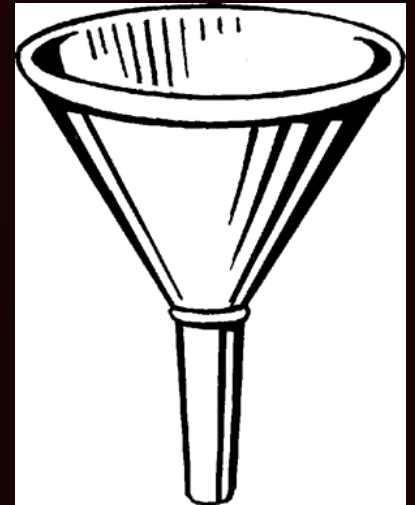
- Muda as permissões de um arquivo
- Modo de uso:
 - `chmod <octais> <arquivo>`
- Exemplo:
 - `chmod 755 file`
- Outra opção de uso:
 - `chmod g+r file` (adiciona a permissão de leitura para o grupo)
- Só o dono do arquivo (ou o usuário root) pode mudar as permissões



Alguns comandos básicos

- grep

- Busca um padrão de texto em um arquivo, retorna as linhas que casaram com esse padrão
- Modo de uso:
 - `grep <padrao> [arquivo]`
- Se não fornecer um arquivo, lê da entrada padrão
 - Alguns parâmetros úteis:
 - `grep -r`: busca recursivamente nos arquivos a partir do diretório atual
 - `grep -i`: ignora caixa alta/baixa
 - `grep -P`: usa sintaxe de expressão regular do perl



Alguns comando básicos

Muitos outros comandos úteis:

- mv (mover arquivo), cp (copiar arquivo), rm (remover arquivo), mkdir (criar diretório), tar (compactar/descompactar arquivos), history (exibir o histórico de comandos), chown (mudar o dono de um arquivo), find (buscar arquivos), echo (exibir texto na saída padrão), let (avalia expressões aritméticas), less (ler o conteúdo de um arquivo), wc (contar linhas/palavras/letras)...

Caracteres coringa (wildcards)

Caracteres especiais, úteis para referenciar a vários arquivos para um comando (executa várias vezes o mesmo comando para vários arquivos)

- *: casa com qualquer caractere, zero ou mais vezes
 - Exemplo: `cat /var/log/secure*`
- ?: casa com qualquer caractere, exatamente uma vez
 - Exemplo: `cat /var/log/secure.?`
- [1-9] ou [a-z]: casa com qualquer caractere no intervalo, exatamente uma vez
 - Exemplo: `cat /var/log/secure.[2-4]`
- [abcd]: casa com qualquer caractere no conjunto, exatamente uma vez
 - Exemplo: `cat /var/log/secure.[13]`

Editores de texto

Use o Vim!

- Modo de uso: vim [nome do arquivo]
- Comandos básicos:
 - 'i': iniciar a edição
 - ':q': sair do editor
 - ':wq': sair do editor salvando mudanças
 - ':q!' sair do editor ignorando mudanças
 - 'u': desfaz última ação
 - ctrl + 'r': refaz última ação
 - '/palavra' + enter: busca palavra no texto
 - '/%s/palavra/outrapalavra': substitui todas as ocorrências de 'palavra' por 'outrapalavra' em todo o texto



Criação de scripts

É possível criar scripts em bash, e executá-los a partir da linha de comando. Existem inclusive instruções condicionais e loops. Exemplos:

```
#!/bin/sh
# isso é um comentário
if [ $UID -eq 0 ] #precisa dos espaços dentro
                  #dos colchetes!
then
    echo "Ola root!"
fi
```



Criação de scripts

```
#!/bin/bash
declare -A array
x=0
# vejamos 'man test' para entender as
# instruções condicionais
while [ $x -lt 10 ]; do
    array[$x]=$RANDOM
    let x++
done
x=0
echo "gerei um array randomico:"
while [ $x -lt 10 ]; do
    echo ${array[$x]}
    let x++
done
```



Pipe e redirecionamento de I/O

Podemos manipular a entrada/saída dos programas através de alguns caracteres especiais:

- '>': redireciona a saída padrão de um programa para outro arquivo
 - Exemplo: `grep GET /var/log/httpd/access_log > get.txt`
- '<': redireciona um arquivo para a entrada padrão de um programa:
 - Exemplo: `/bin/bash < script.sh`
- '2>': redireciona a saída de erros de um programa para um arquivo:
 - Exemplo: `make 2> errors.txt`
 -
 -

Pipe e redirecionamento de I/O

- '|': Redireciona a saída padrão de um programa para a entrada padrão de outro
 - Útil para combinar vários comandos!
 - Exemplo:
 - `grep GET access_log | wc -l`



Gerenciando processos

Alguns comandos úteis para gerenciar processos:

- `ps [opções]`
 - Exibe informações dos processos (pid, usuário, consumo de cpu/memória, etc..) rodando no sistema.
 - Três sintaxes diferentes para as opções (UNIX, BSD, GNU)
 - Exemplos:
 - `ps aux`: lista todos os processos do sistema
 - Usem a criatividade para filtrar a saída do `ps` com pipes

Gerenciando processos

- nice

- Estabelece o 'nível de niceness' do processo que será executado
- Modo de uso: `nice -n <nível> <comando>`
- O nível varia de -20 (maior preferência no escalonamento) a 19 (menor preferência)
- Um usuário comum não pode colocar um nível menor (de maior preferência) do que niceness default

Gerenciando processos

- renice

- Ajusta o 'nível de niceness' de um processo que está em execução
- Modo de uso: renice <nível> <pid do processo>



Gerenciando processos

- kill

- Envia um sinal para um processo encerrar
- O programa em execução pode saber lidar com o sinal e encerrar corretamente, se não souber lidar, morre!
- Modo de uso: `kill [-s <sinal>] pid`
- O sinal padrão é o TERM. Este sinal pode ser tratado pelo programa, portanto pode não encerrá-lo
- O sinal KILL não pode ser tratado, portanto mata o programa de qualquer forma



Gerenciando processos

- top

- Exibe uma interface interativa e em tempo real para a administração dos processos em execução
- Alguns comandos:
 - 'h': exibe um menu de ajuda
 - 'q': sai do programa
 - 'k': mata um processo
 - 'r': renice de um processo



Logs

- Syslog
- Vários arquivos no /var/log:
 - messages: mensagens do sistema e programas
 - syslog: mensagens do kernel
 - secure: login, su, passwd
 - cron
 - maillog
 - debug
 - spooler: news e uucp
- /etc/syslog.conf



Logs

- dmesg
- lastlog
- last
- logger:

\$ logger -s -p user.debug Macacos me mordam, Batman!



Logs

- logrotate:
 - logfile
 - logfile.1
 - ...
 - logfile.n -> mais velho
- Cuidado: coisas velhas são descartadas!
- Evita entupir o disco
- /etc/logrotate.conf



Tarefas

- Criar um usuario na sua máquina e configurar o desktop
- Se familiarizar com o sistema
- Ler e entender a apostila:
<http://piterpunk.unitednerds.org/artigos/aposhell.ps>

