Maringá-PR

Apostila Linux

2012

Introdução

O principal objetivo desta apostila é apresentar um pouco da história do Linux os seus recursos e como usar comandos básicos e úteis do dia-a-dia. Espero que consigam aprender e aplicar bastante o que será explicado aqui além de tirar de uma vez por todas esse paradoxo de que Linux é difícil de usar.

Bons estudos e boa sorte! :)

Autor: Carlos Demetrio

História

A História do Linux começou em 1991 com o início de um projeto pessoal de um estudante Finlandês chamado **Linus Torvalds** de criar uma novo núcleo (*kernel*) de sistema operacional. Desde então, o núcleo Linux resultante foi marcado por um crescimento constante através de sua história. A partir do lançamento inicial de seu **código-fonte**, cresceu de um pequeno grupo de arquivo em **C** sob uma proibitiva licença de distribuição comercial para em 2012, possuir mais de 400 Mb. de fonte sob a licença **GPL**.

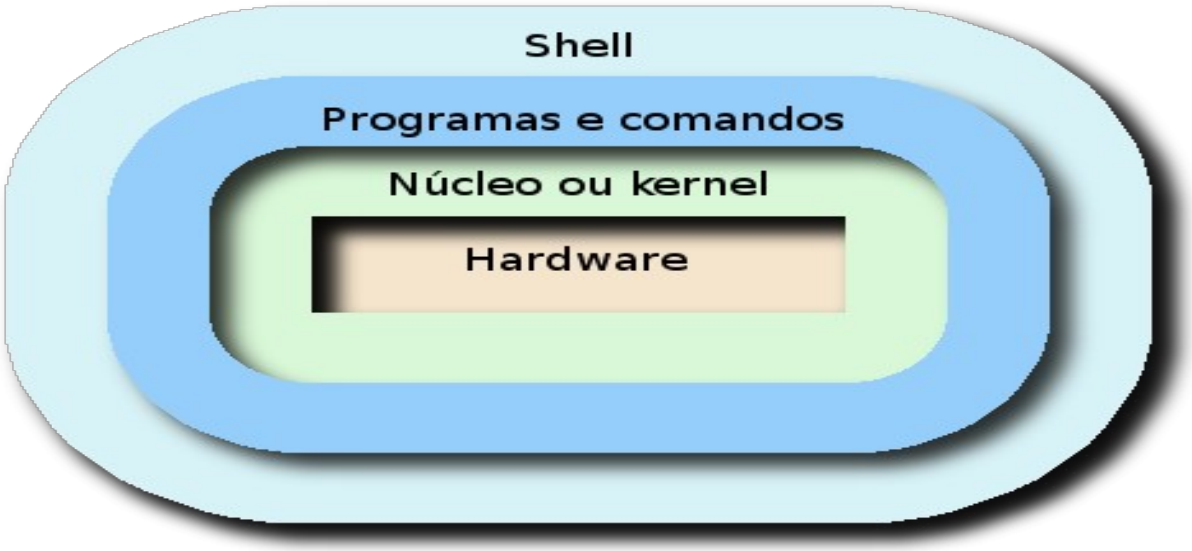
O que é kernel?

Kernel pode ser entendido como o núcleo do sistema operacional, isto é, como a parte essencial deste. Cabe ao ***kernel*** fazer o intermédio entre o *hardware* e os programas executados pelo computador. Isso significa que a junção do ***kernel*** mais os *softwares* que tornam o computador usável (drivers, protocolos de comunicação, entre outros), de acordo com a sua aplicação, é que formam o sistema operacional em si.

Para compreender melhor, você pode imaginar o kernel como sendo o chassi de um veículo. De acordo com a aplicação em questão, uma montadora pode adquirir um chassi e utilizá-lo para montar um carro para transportar cargas ou, se a necessidade for esta, para construir um automóvel de passeio para uma família.

Na data de Setembro de **1991**, a **versão 0.01 do Linux** foi lançada e enviada para o servidor **FTP** (ftp.funet.fi) da *Universidade de Helsinki de Tecnologia* (em Inglês: **HUT**), possuía **10.239 linhas de código**. Atualmente a **versão 3** do Linux foi lançada em 22 de Julho de 2011 em comemoração aos **20 anos do Linux**, este núcleo foi lançado com **13.872.245 linhas de código**. O núcleo do Linux a princípio só mudava o primeiro número da versão em caso de mudanças drásticas, mas para comemorar os 20 anos do Linux e reduzir os números de controles de versão, Torvalds resolver mudar para a versão 3 mesmo sem mudanças significativas no núcleo do Linux. Em Outubro de 2011 é lançada a versão 3.1 do núcleo Linux e em **03 de Janeiro de 2012** é apresentada ao mundo a **versão 3.2.1** do núcleo do Linux com **14.304.901 linhas de código**.

Aplicação da teória:



Distribuições do Linux

O que é uma distribuição?

Distribuição Linux é um Sistema Operacional *Unix-Like* incluindo o *kernel Linux* e outros softwares de aplicação, formando um conjunto. Distribuições (ou "distros") mantidas por organizações comerciais, como a **Red Hat, Ubuntu, SUSE** e **Mandriva**, bem como projetos comunitários como **Debian** e **Gentoo** montam e testam seus conjuntos de software antes de disponibilizá-los ao público.

Como o Linux e a maior parte dos softwares incluídos em distribuições são livres, qualquer organização ou indivíduo suficientemente motivado podem criar e disponibilizar (comercialmente ou não) a sua própria distribuição. Isso faz com que hoje haja registro de mais de **300 distribuições** ativamente mantidas, embora **menos de 10** delas sejam mesmo largamente conhecidas. Quase todas tem ***live cd*** (roda sem instalar).

O Linux evolui muito rapidamente, e os principais distribuidores tendem a lançar versões novas a cada 3 ou 4 meses, ou pelo menos semestralmente. Como em geral você pode obter o software gratuitamente ou a custo baixíssimo, não faz sentido optar pela versão antiga.

Segue uma lista parcial de distribuições de Linux:

Kurumin (Brasileira – Carlos Morimoto)
Debian
Fedora
Gentoo
Mandriva (Mandrake[França] e Conectiva[Brasileira])
Red Hat (CentOS)
Slackware
SUSE
Ubuntu

Como saber qual versão e distribuição do Linux estou usando?

Existe alguns comandos/ arquivos que contém essas informações.

O comando mais usado, por estar em quase todas as versões do Linux é o **uname**:
<prompt> uname -a
Linux demetrio-HP-G42-Notebook-PC 3.0.0-16-generic #28-Ubuntu SMP Fri Jan 27 17:50:54 UTC 2012 i686 i686 i386 GNU/Linux

Você também pode use se tiver o comando que mostra essa informação mais detalhada no Linux é o **lsb_release**:
<prompt> lsb_release -a
Distributor ID: LinuxMint
Description: Linux Mint 12 Lisa
Release: 12
Codename: lisa

Arquivos: */proc/version, /etc/issue, etc*. Tem informações sobre a versão e distribuição.

Shell

O que é shell?

O **shell** é um módulo que atua como interface Usuário/Sistema Operacional,possuindo diversos comandos internos que permitem ao usuário solicitar serviços do S.O. O *shell* também implementa um linguagem simples de programação que permite o desenvolvimento de pequenos programas (os famosos shell scripts). O *shell* mais famoso de Linux é o **Bash**, pois o mesmo oferece vários recursos que facilitam a vida do usuário (ex.: auto-completar).

O usuário normal no shell aparece com o símbolo $, já para o root o símbolo é o #.

Os comandos não embutidos são programas que se iniciam invocando-se um arquivo executável em algum lugar no sistema de arquivos do Linux (o *shell* pesquisa em todos os diretórios listados na variável de ambiente **PATH**). **

Os arquivos carregados pelo shell definem as **variáveis de ambiente**, que nada mais são que definições e valores que o shell e os outros programas do sistema reconhecem. Para ver quais as variáveis de ambiente no seu sistema você pode digitar **printenv**, **env** ou **set**. Por exemplo, são algumas das variáveis de ambiente do **bash**:

$ Mostra o número do processo do comando em execução (PID).
SHELL Mostra o nome do shell atualmente em uso.
PATH Mostra caminho de busca dos comandos digitados pelo usuário.
HOME Mostra o diretório home do usuário.
OLDPWD Mostra o diretório anterior de trabalho do usuário
PWD Mostra o diretório atual de trabalho do usuário.
PS1 Mostra a definição do prompt da linha de comando.
LOGNAME Mostra o nome de acesso do usuário.
USER Mostra o nome do usuário atual.
UID Mostra o número de identificação do usuário.
HISTFILE Mostra o nome do arquivo que armazena as linhas de comando digitadas pelo usuário (no shell bash o arquivo padrão é o **.bash_history**).
HISTSIZE Mostra o número de linhas de comando digitadas pelo usuário que são memorizadas pelo sistema.

echo

Basicamente, a função do comando echo é mostrar mensagens na tela. Também é usado para apresentar o conteúdo da váriavel.
echo [mensagem] [váriavel]

Exemplo:
<prompt> echo “Teste de mensagem”
Teste de mensagem

<prompt> echo $SHELL
/bin/bash
<prompt> echo $PATH
/usr/lib/lightdm/lightdm:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

Alguns comandos que serão apresentados aqui fazem referência a essas váriaveis.

Navegando pelos diretórios

ls

Lista os arquivos de um diretório.
ls [opções] [caminho/arquivo] [caminho1/arquivo1] ...
opções:

-a, –all Lista todos os arquivos (inclusive os ocultos) de um diretório.

-d, –directory Lista os nomes dos diretórios ao invés do conteúdo.

-h, –human-readable Mostra o tamanho dos arquivos em Kbytes, Mbytes ou Gbytes.

-R, Lista diretórios e sub-diretórios recursivamente.

-l, Usa o formato longo para listagem de arquivos. Lista as permissões, data de modificação, donos, grupos, etc. Explicação:

Uma listagem feita com o comando **ls -l** normalmente é mostrada da seguinte maneira:

-rw-rw-r-- 1 demetrio demetrio 20 2012-03-10 19:13 teste

Onde:
-rw-rw-r-- São as permissões de acesso ao arquivo teste. A primeira letra (da esquerda) identifica o tipo do arquivo, se tiver um **“d”** é um *diretório*, se tiver um **“-”** é um *arquivo normal*, **“l”** é um *link*, **“s”** é um arquivo *socket*, *etc*.

1 Se for um diretório, mostra a quantidade de sub-diretórios existentes dentro dele. Caso for um arquivo, será 1.

demetrio Nome do dono do arquivo teste.

demetrio Nome do grupo que o arquivo teste pertence.

20 Tamanho do arquivo (em bytes).

2012-03-10 Ano/ Mês/ Dia da criação/ última modificação do arquivo.

19:13 Hora em que o arquivo foi criado/modificado. Se o arquivo foi criado há mais de um ano, em seu lugar é mostrado o ano da criação do arquivo.

teste Nome do arquivo.

Exemplos do uso do comando ls:
• **ls** - Lista os arquivos do diretório atual.
• **ls /bin /lib** - Lista os arquivos do diretório /bin e /lib
• **ls -la /bin** - Listagem completa (vertical) dos arquivos do diretório /bin inclusive os ocultos.

cd

Entra em um diretório. Você precisa ter a permissão de execução para entrar no diretório.
cd [diretório]

diretório: diretório que deseja entrar.

Exemplos:
• Usando **cd** sem parâmetros ou **cd ~**, você retornará ao seu diretório home**($HOME)**.
• **cd /,** retornará ao diretório raíz.
• **cd -,** retornará ao diretório anteriormente acessado**($PWDLD)**.
• **cd ..,** sobe um diretório.
• **cd ../[diretório],** sobe um diretório e entra imediatamente no próximo (por exemplo, quando você está em /mnt/local, você digita cd ../teste, o comando cd retorna um diretório (/mnt) eentra imediatamente no diretório teste (/mnt/teste).

pwd

Mostra o nome e caminho do diretório atual **($PWD).**

Você pode usar o comando pwd para verificar em qual diretório se encontra (caso seu **$PS1** não mostre isso).

mkdir

Cria um diretório no sistema. Um diretório é usado para armazenar arquivos
mkdir [opções] [caminho/diretório] [caminho1/diretório1]
opção:

-p, Caso os diretórios dos níveis acima não existam, eles também serão criados.

Exemplo:
mkdir -p /mnt/teste/teste2, cria o diretório teste e dentro o teste2.

rmdir

Remove um diretório vazio do sistema

Manipulando arquivos

cat

Mostra o conteúdo de um arquivo binário ou texto.
cat [opções] [diretório/arquivo] [diretório1/arquivo1]
opções:

-n, –number Mostra o número das linhas enquanto o conteúdo do arquivo é mostrado.

-s, –squeeze-blank Não mostra mais que uma linha em branco entre um parágrafo e outro.

rm

Apaga arquivos. Também pode ser usado para apagar diretórios e sub-diretórios vazios ou que contenham arquivos.
rm [opções][caminho][arquivo/diretório] [caminho1][arquivo1/diretório1]
opções:

-i, –interactive Pergunta antes de remover, esta é ativada por padrão.

-v, –verbose Mostra os arquivos na medida que são removidos.

-r, –recursive Usado para remover arquivos em sub-diretórios. Esta opção também pode ser usada para remover sub-diretórios.

-f, –force Remove os arquivos sem perguntar.

cp

Copia arquivos.
cp [opções] [origem] [destino]
opções:

-a, --archive Preserva permissão total.

i, –interactive Pergunta antes de substituir um arquivo existente.

-f, –force Não pergunta, substitui todos os arquivos caso já exista.

-p, Preserva as permissões.

-r, Copia arquivos dos diretórios e subdiretórios da origem para o destino. É recomendável usar -R ao invés de -r.

-R, –recursive, Copia arquivos e sub-diretórios (como a opção -r) e também os arquivos especiais FIFO e dispositivos.

mv

Move ou renomeia arquivos e diretórios. O processo é semelhante ao do comando cp mas o arquivo de origem é apagado após o término da cópia.
mv [opções] [origem] [destino]
opções:

-f, –force, Substitui o arquivo de destino sem perguntar.

-i, –interactive, Pergunta antes de substituir. É o padrão.

-v, –verbose, Mostra os arquivos que estão sendo movidos.

O comando **mv** copia um arquivo da ORIGEM para o DESTINO (semelhante ao comando **cp**), mas após a cópia o arquivo de ORIGEM é apagado.

touch

Muda a data e hora que um arquivo foi criado. Também pode ser usado para criar arquivos vazios. Caso o touch seja usado com arquivos que não existam, por padrão ele criará estes arquivos.
touch [opções] [arquivos]
Onde:
arquivos: São os arquivos que terão sua data/hora modificados.
opções:

-t MMDDhhmm[AA.ss], Usa Mês (MM), Dias (DD), Horas (hh), minutos (mm) e opcionalmente o Ano (AA) e segundos (ss) para modificação do(s) arquivos ao invés da data e hora atual.

-a, –time=atime, Faz o touch mudar somente a data e hora do acesso ao arquivo.

-c, –no-create, Não cria arquivos vazios, caso os arquivos não existam.

-m, –time=mtime, Faz o touch mudar somente a data e hora da modificação.

-r [arquivo], Usa as horas no [arquivo] como referência ao invés da hora atual.

Exemplos:
• **touch teste** - Cria o arquivo teste caso ele não existir.
• **touch -t 10011230 teste** - Altera da data e hora do arquivo para 01/10 e 12:30.
• **touch -t 120112301999.30 teste** - Altera da data, hora ano, e segundos do arquivo para 01/12/1999 e 12:30:30.
• **touch -t 12011200 *** - Altera a data e hora do arquivo para 01/12 e 12:00.

Controle de Usuários

adduser

Adiciona um usuário ou grupo no sistema. Por padrão, quando um novo usuário é adicionado, é criado um grupo com o mesmo nome do usuário. Será criado um diretório home com o nome do usuário (a não ser que o novo usuário criado seja um usuário do sistema) e este receberá uma identificação. A identificação do usuário (**UID**) escolhida será a primeira disponível no sistema especificada de acordo com a faixa de **UIDS** de usuários permitidas no arquivo de configuração /etc/adduser.conf., este é o arquivo que contém os padrões para a criação de novos usuários no sistema.

adduser [opções] [usuário/grupo]
Onde:
usuário/grupo: Nome do novo usuário que será adicionado ao sistema.
opções:

-disable-passwd, Não executa o programa passwd para escolher a senha e somente permite o uso da conta após o usuário escolher uma senha.

–force-badname, Desativa a checagem de senhas ruins durante a adição do novo usuário.

Por padrão o adduser checa se a senha pode ser facilmente adivinhada.

passwd

Modifica a parametros e senha de usuário. Um usuário somente pode alterar a senha de sua conta, mas o superusuário (root) pode alterar a senha de qualquer conta de usuário.

passwd [usuário] [opções]
Onde:
usuário: Nome do usuário que terá sua senha alterada.
opções:

-e, Força a expiração de senha para a conta especificada.

-d, Delete a senha do usuário.

userdel

Apaga um usuário do sistema. Quando é usado, este comando apaga todos os dados da conta especificado dos arquivos de contas do sistema.

userdel [-r] [usuário]
Onde:

-r, Apaga também o diretório HOME do usuário.

logname
Mostra seu login **($LOGNAME).**
logname

whoami
Mostra quem está logado **($USER).**
whoami

id
Mostra a **ID**(nome-número) do usuário no sistema **($UID).**
id -u

who
Mostra quem está atualmente conectado no computador/ servidor. Este comando lista os nomes de usuários que estão conectados o terminal e data da conexão.
who [opções]
opções:

-d, –dead, Mostra processos mortos no sistema.

-q, –count, Mostra o total de usuários conectados aos terminais.

last
Mostra uma listagem de entrada e saída de usuários no sistema. São mostrados os seguintes campos na listagem:
Nome do usuário
Terminal onde ocorreu a conexão/desconexão
O hostname (caso a conexão tenha ocorrido remotamente) ou console (caso tenha ocorrido localmente).
A data do login/logout, a hora do login/down se estiver fora do sistema/ still logged in se ainda estiver usando o sistema
Tempo (em Horas:Minutos) que esteve conectado ao sistema.
last [opções]
opções:

-n [num], Mostra [num] linhas. Caso não seja usada, todas as linhas são mostradas.

-a, Mostra o hostname na última coluna.

-x, Mostra as entradas de desligamento do sistema e alterações do nível de execução S.O.

Histórico de comandos

$HISTFILE (No shell bash = ***bash_history***)
Arquivo responsável por gravar uma lista dos comandos digitados pelos usuários, localizado no diretório "home" de cada usuário, nele contém os comandos digitados no shell.
Veremos aqui alguns comandos que utiliza esse arquivo e como executar os comandos digitados sem precisar digitar tudo novamente.

Comandos:

$ history 10
Lista os 10 últimos comandos digitados. Sem número ele mostra todos.

$ history -c
Limpa o arquivo ".bash_history".

$ fc -s
Executa o último comando contido no ".bash_history".

$ fc -s ls
Executa o último comando cujo o nome inicie com "ls".

$!!
Executa o último comando contido no ".bash_history".

$!ls
Executa o último comando cujo o nome inicie com "ls".

$!10
Executa o comando que estiver na 10 linha.

CTRL + r texto
Busca recursivamente no .bash_history pelo texto(comando) digitado.

$HISTSIZE
Mostra o número de comandos que o .bash_history vai armazenar.

Executando programa/ comandos

Executando um comando/programa

Para executar um comando, é necessário que ele tenha permissões de execução(x). O programa/comando é executado e receberá um número de identificação (chamado de PID - Process Identification), este número é útil para identificar o processo no sistema e assim ter um controle sobre sua execução.

$PATH

Path é o caminho de procura dos arquivos/comandos executáveis. O path(caminho) é armazenado na variável de ambiente PATH. Lembrando que você pode ver o conteúdo desta variável com o comando echo $PATH. Por exemplo, o caminho: */usr/local/bin:/usr/bin:/bin:/usr/home/usuario/bin* significa que se você digitar o comando **ls**, o interpretador de comandos iniciará a procura do programa ls no diretório */usr/local/bin*, caso não encontre o arquivo no diretório */usr/local/bin* ele inicia a procura em */usr/bin*, até que encontre o arquivo procurado. Caso o interpretador de comandos chegue até o último diretório do path e não encontre oarquivo/comando digitado, é mostrada a seguinte mensagem:
ls: command not found (comando não encontrado)

Caso um arquivo/comando não esteja localizado em nenhum dos diretórios do path, você deve executa-lo usando um **“./”** na frente do comando.

whereis/ which

Localiza um programa na estrutura de diretórios do path. É muito semelhante ao locate, mas a busca é feita no path do sistema e somente são mostrados arquivos executáveis.

Tipos de Execução de comandos/programas

Um programa pode ser executado de duas formas:
1 Primeiro Plano - Também chamado de *foreground*. Quando você deve esperar o término da execução de um programa para executar um novo comando. Somente é mostrado o aviso de comando após o término de execução do comando/programa.

2 Segundo Plano - Também chamado de *background*. Quando você não precisa esperar o término da execução de um programa para executar um novo comando. Após iniciar um programa em background, é mostrado um número PID (identificação do Processo) e o aviso de comando é novamente mostrado, permitindo o uso normal do sistema. O programa executado em background continua sendo executado internamente. Após ser concluído, o sistema retorna uma mensagem de pronto acompanhado do número PID do processo que terminou. Para iniciar um programa em primeiro plano, basta digitar seu nome normalmente. Para iniciar um programa em segundo plano, acrescente o caracter “&” após o final do comando.OBS: Mesmo que um usuário execute um programa em segundo plano e saia do sistema, o programa continuará sendo executado até que seja concluído ou finalizado pelo usuário queiniciou a execução (ou pelo usuário root).

Exemplo: **find / -name “arquivo” >> resultado &**
O comando será executado em segundo plano e deixará o sistema livre para outras tarefas.

Executando programas em seqüência
Os comandos podem ser executados em seqüência (um após o término do outro) se os separarmos com **“;”** (ponto vírgula).
Por exemplo:
clear; ls

ps
Algumas vezes é útil ver quais processos estão sendo executados no computador. O comando ps faz isto, e também nos mostra qual usuário executou o programa, hora que o processo foi iniciado, etc.
ps [opções]
Onde:
opções:

a, Mostra os processos criados por você e de outros usuários do sistema.

x, Mostra processos que não são controlados pelo terminal.

u, Mostra o nome de usuário que iniciou o processo e hora em que o processo foi iniciado.

m, Mostra a memória ocupada por cada processo em execução.

f, Mostra a árvore de execução de comandos (comandos que são chamados por outros comandos).

e, Mostra variáveis de ambiente no momento da inicialização do processo.

w, Mostra a continuação da linha atual na próxima linha ao invés de cortar o restante que não couber na tela.

As opções acima podem ser combinadas para resultar em uma listagem mais completa. Você também pode usar pipes **“|”** para filtrar a saída do comando ps. Ao contrário de outros comandos, o comando **ps não precisa do hífen “-”** para especificar os comandos. Isto porque ele não utiliza opções longas e não usa parâmetros.

Exemplos: **ps, ps aux | grep nome_programa, ps auxf, ps auxw.**

fuser
Idica qual o **PID** do comando/ programa que está usando certo *arquivo* ou *socket*.
fuser [opções]
Onde:
opções:
-k, Mata o processo(s) que está usando o arquivo.(KILL)

-u, --user, Mostra o nome do usuário que está usando o arquivo.

Exemplos: **fuser arquivo, fuser -k arquivo, fuser -u arquivo**

kill

Permite enviar um sinal a um comando/programa. Caso seja usado sem parâmetros, o kill enviará um sinal de término ao processo sendo executado.

kill [sinal] [número]
Onde:
sinal:

-9, Envia um sinal de destruição ao processo ou programa. Ele é terminado imediatamente sem chances de salvar os dados ou apagar os arquivos temporários criados por ele. Você precisa ser o dono do processo ou o usuário root para termina-lo ou destruí-lo.
número É o número de identificação do processo PID.

Exemplo: **kill 500, kill -9 500**

killall

Permite finalizar processos através do nome.

killall [opções] [sinal] [processo]
Onde:
processo: Nome do processo que deseja finalizar
sinal: É o número do sinal que será enviado ao processo.
opções:

-i, Pede confirmação sobre a finalização do processo.

-l, Lista o nome de todos os sinais conhecidos.

-q, Ignora a existência do processo.

-v, Retorna se o sinal foi enviado com sucesso ao processo.

-w, Finaliza a execução do killall somente após finalizar todos os processos.

Exemplo: **killall -i nome_programa**

Eliminando caracteres estranhos

reset

As vezes quando um programa mal comportado é finalizado ou quando você visualiza um arquivo binário através do comando cat, é possível que o aviso de comando (prompt) volte com caracteres estranhos. Para fazer tudo voltar ao normal, basta digitar **reset** e teclar ENTER. Não se preocupe, o comando reset não reiniciará seu computador (como o botão reset do seu computador faz), ele apenas fará tudo voltar ao normal. Note que enquanto você digitar reset aparecerão caracteres estranhos ao invés das letras. Não se preocupe! Basta digitar corretamente e bater ENTER e o aviso de comando voltará ao normal.

Comandos diversos

Comandos de uso diversos no sistema.

clear

Limpa a tela e posiciona o cursor no canto superior esquerdo do vídeo.
clear

date

Permite ver/modificar a Data e Hora do Sistema. Você precisa estar como usuário root para modificar a data e hora.

date MesDiaHoraMinuto[AnoSegundos]
Onde:
MesDiaHoraMinuto[AnoSegundos] São respectivamente os números do mês, dia, hora e minutos sem espaços. Opcionalmente você pode especificar o Ano (com 2 ou 4 dígitos) e os segundos.

Se corrigir a hora do sistemas, deve-se verificar a “data” e “hora” da **BIOS**. Para isso usa o comando hwclock. Caso a data/hora esteja errada usa se o comando: **hwclock -w**, e se for fazer o processo **inverso** colocar os dados da **BIOS** para o sistema usa o **hwclock -s**.

cal

O comando **cal** é utilizado para mostrar um calendário de um mês e/ou ano em específico. Onde mês é o numero do mês (01 a 12) e ano, é o ano no formato aaaa (4 algarismos).

Exemplos: **cal**

March 2012

Su	*Mo*	*Tu*	*We*	*Th*	*Fr*	*Sa*
				1	*2*	*3*
4	*5*	*6*	*7*	*8*	*9*	*10*
11	*12*	*13*	*14*	*15*	*16*	*17*
18	*19*	*20*	*21*	*22*	*23*	*24*
25	*26*	*27*	*28*	*29*	*30*	*31*

Teste: **cal MM AAAA** (do seu nascimento)
Obs.: **MM** = Mês e **AAAA** = Ano.

df

Mostra o espaço livre/ocupado de cada partição.
df [opções]
onde:
opções:
-h, –human-readable, Mostra o espaço livre/ocupado em MB, KB, GB ao invés de blocos.

-H, Idêntico a **-h** mas usa **1000** ao **invés** de **1024** como unidade de cálculo.

-l, Somente lista sistema de arquivos locais.

-m, mostra em MB

Exemplos: **df, df -h, df -l.**

du

Mostra o espaço ocupado por arquivos e sub-diretórios do diretório atual.
du [opções]
onde:
opções:
-a, –all, Mostra o espaço ocupado por todos os arquivos.

-D, Não conta links simbólicos.

-h, –human, Mostra o espaço ocupado em formato legível por humanos (Kb, Mb) ao invés de usar blocos.

-H, Como o anterior mas usa 1000 e não 1024 como unidade de cálculo.

-m, Mostra o espaço ocupado em Mbytes.

-S, -s, –separate-dirs, Não calcula o espaço ocupado por sub-diretórios.

Exemplo: **du -h, du -sh.**

free

Mostra detalhes sobre a utilização da memória RAM do sistema.
free [opções]
Onde:
opções:
-m, Mostra o resultado em Mbytes.

-o, Oculta a linha de buffers.

O **free** é uma interface ao arquivo */proc/meminfo.*

Exemplo: **free -m, free -mo**

grep

Procura por um texto dentro de um arquivo(s) ou no dispositivo de entrada.
grep [expressão] [arquivo]
Onde:
expressão: É a palavra ou frase que será procurada no texto. Se tiver mais de 2 palavras você deve identifica-la com aspas “ ” caso contrário o grep assumirá que a segunda palavra é o arquivo!
arquivo: É o arquivo onde será feita a procura.
Também usar o comando para filtrar o resultado de uma comando, por exemplo:
ps aux | grep nome_do_programa

Outro exemplo: **grep “capitulo” texto.txt.**

more

Permite fazer a paginação de arquivos ou da entrada padrão. O comando more pode ser usado como comando para leitura de arquivos que ocupem mais de uma tela. Quando todaa tela é ocupada, o more efetua uma pausa e permite que você pressione **Enter** ou **espaço** para continuar avançando no arquivo sendo visualizado. Para sair do more pressione **q**.

more [arquivo]
Onde:
arquivo: É o arquivo que será paginado.

Exemplo: **who | more**

less

Permite fazer a paginação de arquivos ou da entrada padrão. O comando **less** pode ser usado como comando para leitura de arquivos que ocupem mais de uma tela. Quando toda a tela é ocupada, o less efetua uma pausa (semelhante ao more) e permite que você pressione Seta para Cima e Seta para Baixo ou **PgUP/PgDown** para fazer o rolamento da página. Para sair do less pressione **q**.

less [arquivo]
Onde:
arquivo: É o arquivo que será paginado.

Exemplos: **ps aux | less**

uptime

Mostra o tempo de execução do sistema desde que o computador foi ligado. Caso o sistema foi reiniciado o contador também será reiniciado.

Faz referência ao arquivo: */proc/uptime*

su

Permite o usuário mudar sua identidade para outro usuário sem fazer o logout. Útil para executar um programa ou comando como root sem ter que abandonar a seção atual.
su [usuário] [-c comando]
Onde:
usuário: É o nome do usuário que deseja usar para acessar o sistema. Se não digitado, é assumido o usuário root.
-c comando: Caso seja especificado **-c comando**, executa o comando sob o usuário especificado. Será pedida a senha do superusuário para autenticação. Digite ***exit*** quando desejar retornará identificação de usuário anterior.

ln

Cria links para arquivos e diretórios no sistema. O link é um mecanismo que faz referência a outro arquivo ou diretório em outra localização. O link em sistemas GNU/Linux faz referência reais ao arquivo/diretório podendo ser feita cópia do link (será copiado o arquivo alvo), entrar no diretório (caso o link faça referência a um diretório), etc.

ln [opções] [origem] [link]
Onde:
origem: Diretório ou arquivo de onde será feito o link.
link: Nome do link que será criado.
opções:
-s, Cria um link simbólico. Usado para criar ligações com o arquivo/diretório de destino.

-v, Mostra o nome de cada arquivo antes de fazer o link.

ftp

Permite a transferência de arquivos do computador remoto/local e vice versa. O file transfer protocol é o sistema de transmissão de arquivos mais usado na Internet. É requerida a autenticação do usuário para que seja permitida a conexão.
ftp [ip/dns]
Abaixo alguns dos comandos mais usados no **FTP**:

ls, Lista arquivos do diretório atual.
cd [diretório], Entra em um diretório.
get [arquivo], Copia um arquivo do servidor ftp para o computador local. O arquivo é gravado, por padrão, no diretório onde o programa ftp foi executado.
put, Envia um arquivo para o diretório atual do servidor FTP.
send [arquivo], Também envia um arquivo para o diretório atual do servidor FTP.
hash [on/off], Por padrão esta opção está desligada. Quando ligada, faz com que o caracter **“#”** seja impresso na tela indicando o progresso do download.
mget [arquivos], Semelhante ao **get**, mas pode copiar diversos arquivos e permite o uso de curingas.
mput [arquivos], Semelhante ao **put**, mas pode copiar diversos arquivos e permite o uso de curingas.
prompt [on/off], Ativa ou desativa a pergunta para a cópia de arquivo. off = sim para todas

Permissões dos arquivos

chmod

Muda a permissão de acesso a um arquivo ou diretório. Com este comando você pode escolher se usuário ou grupo terá permissões para ler, gravar, executar um arquivo ou arquivos. Sempre que um arquivo é criado, seu dono é o usuário que o criou e seu grupo é o grupo do usuário.
chmod [opções] [permissões] [diretório/arquivo]
Onde:
diretório/arquivo: É o diretório ou arquivo que terá sua permissão mudada.
opções:
-v, –verbose, Mostra todos os arquivos que estão sendo processados.

-f, –silent, Não mostra a maior parte das mensagens de erro.

-c, –change, Semelhante a opção -v, mas só mostra os arquivos que tiveram as permissões alteradas.

-R, –recursive, Muda permissões de acesso do diretório/arquivo no diretório atual e seus sub-diretórios.

• **+-= - +** coloca a permissão, - retira a permissão do arquivo e = define a permissão exatamente como especificado.
• **rwx - r** permissão de read(leitura) do arquivo. w permissão de write(gravação). x permissão de execute(execução) (ou acesso a diretórios).

u = o dono do arquivo;
g = os usuários que são membros do mesmo grupo do arquivo;
o = os usuários que não membros do grupo do arquivo;
a = qualquer usuário do sistema.

Modo Octal

É possível também utilizar o modo octal para alterar as permissões de acesso a um arquivo. Abaixo mostramos a lista das permissões de acesso no ***modo octal***.

Valor Octal	Valor Binário rwx	Significado
0	000	nenhuma permissão de acesso
1	001	permissão de execução (x)
2	010	permissão de gravação (w)
3	011	permissão de gravação e execução(wx)
4	100	permissão de leitura (r)
5	101	permissão de leitura e execução (rx)
6	110	permissão de leitura e gravação (rw)
7	111	permissão de leitura, gravação e execução (rwx)

chown

Muda dono de um arquivo/diretório. Opcionalmente pode também ser usado para mudar o grupo.

chown [opções] [dono.grupo] [diretório/arquivo]
onde:
dono.grupo: Nome do dono.grupo que será atribuído ao diretório/arquivo. O grupo é opcional.
diretório/arquivo Diretório/arquivo que o dono.grupo será modificado.
Opções:

-v, –verbose, Mostra os arquivos enquanto são alterados.

-f, –supress, Não mostra mensagens de erro durante a execução do programa.

-c, –changes, Mostra somente arquivos que forem alterados.

-R, –recursive, Altera dono e grupo de arquivos no diretório atual e sub-diretórios.
O dono.grupo pode ser especificado usando o nome de grupo ou o código numérico correspondente.

 Pesquise sobre o arquivo: */etc/passwd*

umask

A **umask** (*user mask*) são **3 números** que definem as permissões iniciais do dono, grupo e outros usuários que o arquivo/diretório receberá quando for criado ou copiado para um novo local. Digite **umask** sem parâmetros para retornar o valor de sua **umask** atual.

Valor (permissão...)
6 sem permissão.
5 Permissão apenas para executar.
4 Permissão apenas para gravar.
3 Permissão para gravar e executar.
2 Permissão apenas para ler.1 Permissão para executar e ler.
0 Completo – leitura / execução / escrita.

Exemplo:
demetrio@demetrio-HP-G42-Notebook-PC ~ $ umask
0002
demetrio@demetrio-HP-G42-Notebook-PC ~ $ touch teste
demetrio@demetrio-HP-G42-Notebook-PC ~ $ ls -la teste
-rw-rw-r-- 1 demetrio demetrio 0 2012-03-11 13:11 teste
demetrio@demetrio-HP-G42-Notebook-PC ~ $ chmod 777 teste
demetrio@demetrio-HP-G42-Notebook-PC ~ $ chown demetrio.demetrio teste
demetrio@demetrio-HP-G42-Notebook-PC ~ $ ls -la teste
-rwxrwxrwx 1 demetrio demetrio 0 2012-03-11 13:11 teste

alias

Tradução livre é “apelido”. Esse comando é usado para combinar um ou mais comando com parâmetros para um outro nome teoricamente mais fácil de lembrar.

Exemplo:
Se quiser transformar o comando ls em um comando chamado **listar** e usar nele outros parametros que você geralmente usa com esse comando é só executar o comando:

alias listar='ls -lat'

Para mostras os “***alias***” que existe no seu sistema basta executar o comando sem parâmetros. **;)**

Entendendo redirecionameto e PIPE:

>

Redireciona a saída padrão de um programa/comando/script para algum dispositivo ou arquivo ao invés do dispositivo de saída padrão (tela). Quando é usado com arquivos, este redirecionamento cria ou substitui o conteúdo do arquivo.

>>

Redireciona a saída padrão de um programa/comando/script para algum dispositivo ou adiciona as linhas ao final de arquivo ao invés do dispositivo de saída padrão (tela). A diferença entre este redirecionamento duplo e o simples, é se caso for usado com arquivos, adiciona a saída do comando ao final do arquivo existente ao invés de substituir seu conteúdo.

<

Direciona a entrada padrão de arquivo/dispositivo para um comando. Este comando faz o contrário do anterior, ele envia dados ao comando.

<<

Este redirecionamento serve principalmente para marcar o fim de exibição de um bloco. Este é especialmente usado em conjunto com o comando cat. Exemplo:

cat << final
este arquivo será mostrado até que a palavra final seja
localizada no inicio da linha
final

| (pipe)

Envia a saída de um comando para a entrada do próximo comando para continuidade do processamento. Os dados enviados são processados pelo próximo comando que mostrará o resultado do processamento.

A principal diferença entre o “|” e o “>”, é que o Pipe envolve processamento entre comandos, ou seja, a saída de um comando é enviado a entrada do próximo e o “>” redireciona a saída de um comando para um arquivo/dispositivo.

Como obter ajuda do sistema?

As páginas de manual acompanham quase todos os programas **GNU/Linux**. Elas trazem uma descrição básica do comando/programa e detalhes sobre o funcionamento de opção. Uma página de manual é visualizada na forma de texto único com rolagem vertical. Também documenta parâmetros usados em alguns arquivos de configuração.

man

A utilização da página de manual é simples, digite:
man [seção] [comando/arquivo]
onde:
seção: É a seção de manual que será aberta, se omitido, mostra a primeira seção sobre o comando encontrado (em ordem crescente).
comando/arquivo: Comando/arquivo que deseja pesquisar.
A navegação dentro das páginas de manual é feita usando-se as teclas:
• **q** - Sai da página de manual
• **PageDown ou f** - Rola 25 linhas abaixo
• **PageUP ou w** - Rola 25 linhas acima
• **SetaAcima ou k** - Rola 1 linha acima
• **SetaAbaixo ou e** - Rola 1 linha abaixo
• **r** - Redesenha a tela (refresh)
• **p ou g** - Inicio da página
• **h** - Ajuda sobre as opções da página de man

--help

Ajuda rápida, é útil para sabermos quais opções podem ser usadas com o comando/programa. Quase todos os comandos/programas **GNU/Linux** oferecem este recurso que é útil para consultas rápidas (e quando não precisamos dos detalhes das páginas de manual). É útil quando se sabe o nome do programa mas deseja saber quais são as opções disponíveis e para o que cada uma serve. Para acionar o ***help on line***, digite:
[comando] –**help**
onde:
comando: É o comando/programa que desejamos ter uma explicação rápida.

apropos/whatis

O **apropos** procura por programas/comandos através da descrição. É útil quando precisamosfazer alguma coisa mas não sabemos qual comando usar. Para usar o comando ***apropos*** digite:
apropos [descrição]

O **whatis** explica o que o comando setado faz.
Para usar digite:
whatis [comando]

Exemplo: **apropos clear, whatis clear**

Serviços

Os serviços de rede são inicializados com o *Linux* no momento do *boot* estão intimamente relacionados ao *software* em que estão instalado. Nome de alguns serviços:

sshd
smb
network
mysqld

Opções:

status Situação do serviço
start Inicia o serviço.
stop Para o serviço.
restart Para e inicia novamente o serviço.
Reload Reinicia sem parar o serviço***.***

SSH

O ***Secure Shell*** ou **SSH** é, simultaneamente, um programa e um protocolo de rede que permite a conexão com outro computador na rede, de forma a executar comandos de uma unidade remota. Possui as mesmas funcionalidades do **TELNET**, com a vantagem da conexão entre o cliente e o servidor serem criptografadas. O arquivo de configuração é */etc/ssh/sshd_config.* Alguns parametros importantes:

Port 22	Porta de comunicação
PermitRootLogin yes	Permite ou não o login direto do root.
PermitEmptyPasswords no	Login com usuário sem senha (passwd -d user).

Serviço é o **sshd**, se alterar alguns desses parâmetros tem que reiniciar o mesmo:

service sshd restart

SAMBA

O ***Samba*** surgiu, basicamente, pela combinação de três fatores:

- Necessidade de compartilhamento;
- Linux vs Windows;
- Redes de Computadores atuais.

O *Linux* vem se firmando a cada dia como um sistema operacional robusto e seguro, bem como largamente utilizado em servidores de rede (HTTP, E-mail, NFS, Firewall, DHCP, NFS, o próprio SMB, etc). Já o Microsoft Windows é um sistema operacional popular, utilizado em cerca de 95% dos computadores pessoais no mundo. Surgiram então as redes locais (LAN - Local Área Network), que são extremamente baratas e confiáveis - e com elas a necessidade do compartilhamento de arquivos e impressoras. É justamente para unir as três tecnologias - Linux, Windows e Redes - com a necessidade de compartilhamento que

surgiu o Samba. O Samba é um aplicativo servidor (server side), ou seja, é executado no servidor (aqui, Linux, mas também pode ser utilizado em UNIXes em geral. Também já foi portado para uma série de plataformas não-UNIX como Novell NetWare por exemplo). Mesmo rodando em uma plataforma completamente diferente à plataforma Windows, o Samba se comporta como se fosse tal plataforma. Mais precisamente, o Samba "conversa" com o Windows como se fosse o tal.

Para entender melhor esta integração do Samba às Redes Windows, devemos focar agora nossa atenção ao Ambiente de Rede Windows. Neste ambiente que se encontram todos os computadores Windows, e é nele que se concentram os compartilhamentos de arquivos e impressoras. Então, o Samba nasceu para fazer com que os computadores operando sobre o UNIX participassem deste Ambiente de Rede sem que ocorram conflitos. Um usuário Windows utiliza arquivos ou impressoras deste servidor Linux como se tal fosse realmente outro computador com plataforma Windows.

Toda essa gama de soluções é gerenciada pelo protocolo ***CIFS*** (Common Internet File System). É neste protocolo que opera o também protocolo **SMB** (Server Message Block), que deu origem então, há alguns anos atrás - devido a uma necessidade particular de compartilhamento de impressora - ao Samba, este que é uma implementação do CIFS, totalmente Open Source (Código Aberto).

O arquivo de configuração é
/etc/samba/smb.config. Exemplo alguns parâmetros:

```
[global]
# Nome do grupo do ambiente Windows
workgroup = nome_da_rede
# Exige login/ senha
security = user
public = no
[compartilhamento]
comment = Compartilhamento
path = /mnt
```

Serviço é o **smb**, se alterar alguns desses parâmetros tem que reiniciar o mesmo:

```
service smb restart
```

Para adionar usuário no ***samba***, deve:

```
adduser usuario
```

```
# smbpasswd -a usuario
```

E para excluir:

```
# smbpasswd -x usuario
```

Instalando programas na distribuições Red Had(CentOS)

yum

O y**um** é um acrônimo para ***Y****ellow dog* ***U****pdater,* ***M****odified*. É uma ferramenta utilizada para gerenciar a instalação e remoção de pacotes em distribuições Linux, que utilizam o sistema **RPM**. É um gerenciador de pacotes de arquivos similar ao **APT(Debian .deb)** que lida automaticamente com dependências computando-as e resolvendo o que deve ser feito para tratá-las. Trabalha com formato .rpm de pacotes de arquivos. O ***yum*** faz o *download* do pacote especificado de algum repositório. Possui um simples arquivo de configuração. Faz um cálculo eficaz das dependências.

Opções:

install Instala um pacote.
remove Remove um pacote***.***
search Procura por um pacote.
update Atualiza as dependências.
upgrade Faz a atualização de todos os pacotes disponiveis.

Exemplo:
yum install vim (instala o VIM)

RPM

O **RPM**, originalmente abreviatura de ***RedHat Package Manager***, e atualmente um acrónimo recursivo de *RPM Package Manager* (“Gerenciador de Pacotes RPM”) é um sistema de gerenciamento de pacotes de *softwares*, assim como o formato de arquivo usado por esse sistema. O **RPM** serve para instalar, atualizar, desinstalar, verificar e procurar *softwares*. Originalmente desenvolvido pela RedHat, RPM é agora usado por muitas distribuições.

Opções:
-i, instala um pacote
-e, exclui um pacote
-h, exibe a *hast*
-qi, mostra informações sobre pacote.
-qa, consulta pacotes instalados
-v, mostra o status do pacotes

Exemplo:
Para instalar use a seguinte combinação de parâmetros

rpm -ivh pacote.rpm

VIM

O **VIM** é um editor de texto que tem vários recursos e por esse motivo vai ser melhor apresentado aqui suas funções.

No vim temos vários **"modos"**, que são estados do editor.
São eles:

Modo	Tecla	Rodapé	Descrição
de Inserção	i	-- INSERT --	Inserção de texto
de Comandos		<ESC>	Comandos de manipulação de texto
Linha de comando	:	:	Comandos de manipulação arquivo
Visual	v	-- VISUAL --	Seleção visual de texto
Busca	/	/	Busca de padrões no texto
Reposição	R	-- REPLACE --	Inserção sobreescrevendo

Chamando o vim:

vim	Abre o vim vazio, sem arquivo (muito pouco usado)
vim arquivo	Abre o arquivo "arquivo". Se ele não existir, o cria
vim arquivo +	Abre com cursor no fim do arquivo
vim arquivo +10	Abre com cursor na linha 10
vim arquivo +/linux	Abre com cursor na 1ª ocorrência de "linux"

Salvando e saindo:

:w	Salva
:q	Sai
:wq ou :x ou ZZ	Salva e sai
:w!	Salva forçado
:q!	Sai forçado
:wq!	Salva e sai forçado

Obs.: Antes de executar os comandos seguintes, aperte **<ESC>** para ir ao modo de comandos, e é claro, após, um **<ENTER>**

Editando:
Ao entrar no vim, ele está no modo de comandos. Como saber? Olhe no rodapé da tela. Nada na última linha. Para começar a Inserir um texto, aperte **"I"** ou **<INS>**. Você verá que aparecerá um **"-- INSERT --"** no rodapé. Agora você pode digitar seu texto normalmente. Quer parar para dar uma gravada? Veja o tópico acima. **(<ESC>:w)**
chega de editar e quer salvar e sair? Veja acima de novo**. (<ESC>:wq)**

Copiando e colando:

Usando o mouse → **<ESC>:set mouse=a**

No modo de Inserção mesmo, note que o cursor do teclado (esse tracinho piscando na tela), está numa posição, e o cursor do mouse (mexa o mouse para que ele se mova) está em outra posição. Selecione uma parte do texto com o mouse, segurando o botão esquerdo. Ao colar, o texto selecionado com o **MOUSE** será colocado a partir do
cursor do **TECLADO**. Experimente. Dependendo do mouse, a colagem se faz:
- Apertando o botão direito
- Segurando o botão esquerdo e apertando o direito ao mesmo tempo
- **<SHIFT> <INS>**

Tente, o seu será umas das 3 alternativas acima.

Usando o modo **visual**:
Entre no modo visual **(<ESC>v),** e simplesmente aperte as setas do teclado, movendo o cursor do **TECLADO** e selecionando o texto desejado. O comando para copiar o texto é o **"y"** de ***yank***. Volte para o modo de comandos **(<ESC>)** e posicione o cursor do **TECLADO** no lugar onde você quer colar o texto selecionado. O comando de colagem é o **"p"** de p***aste***.

Apagando (delete):

Como no tópico anterior, use o modo visual **(<ESC>v)** para selecionar o texto desejado. Para apagá-lo, digite **"d"**, de ***delete***. Com o linux no modo texto, a tecla **<DEL>** do teclado também funciona para apagar texto.

Outros comandos úteis:

Modo de comando

:5 Vai para a 5 linha.

dd Apaga a linha do cursor.

5dd Apaga as 5 próximas linhas apartir do cursor.

G Vai para o final do arquivo.

gg Vai para o inicio do arquivo.

Se gostou do **VIM**, pesquise sobre outras funções. Você não vai se arrepender. **:)**

Instalando o Linux RedHad – CentOS na Virtual Box

Baixando o software

O VirtualBox é free e open source. Para baixar o software (aproximadamente 90 MB) acesse o site www.virtualbox.org e clique na sessão Downloads. (**Versão atual: 4.1.8**)

Instalando

A instalação do VirtualBox é extremamente fácil. Basta seguir o processo Next, Next e Finish.

Criando uma máquina virtual

Na primeira execução do software será solicitado seu nome e e-mail. Preencha os dados e tique a opção para não ser contatado.

Clique no botão **novo:**

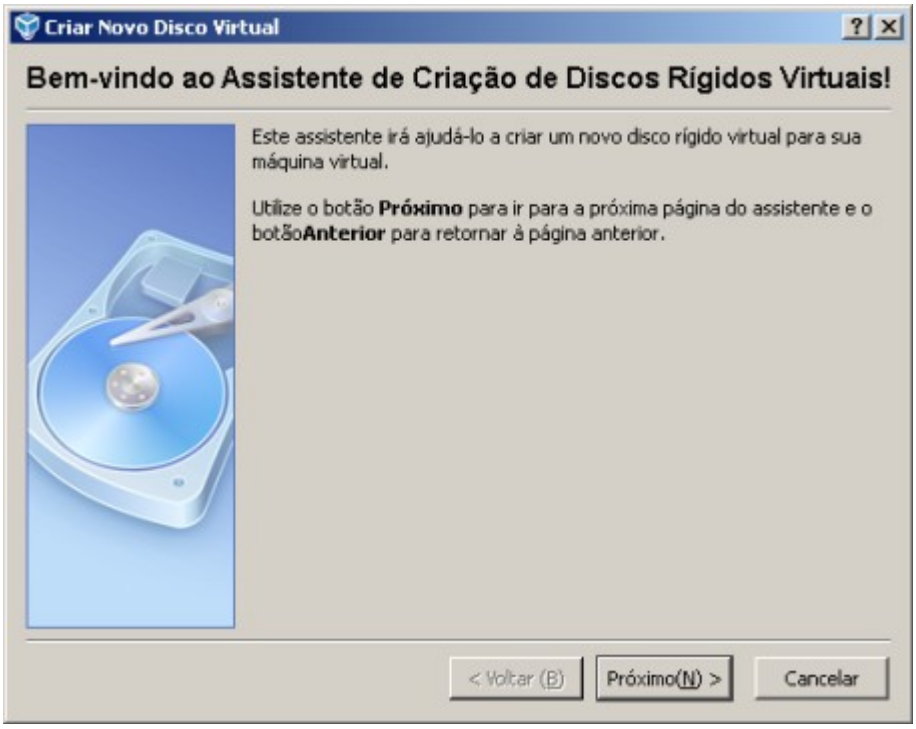
Será aberto um assistente que o ajudará nesta tarefa, clique em **Próximo** para continuar:

Nomeie a máquina virtual e escolha qual será o sistema operacional que ela conterá e clique em **Próximo** para continuar:

Digite agora a quantidade de memória que a máquina virtual irá ter e clique em **Próximo** para continuar:

Na tela seguinte será para você escolher o disco que será usado para a máquina virtual. Neste instante não há disco criado, então clique no botão **Novo** para criar:

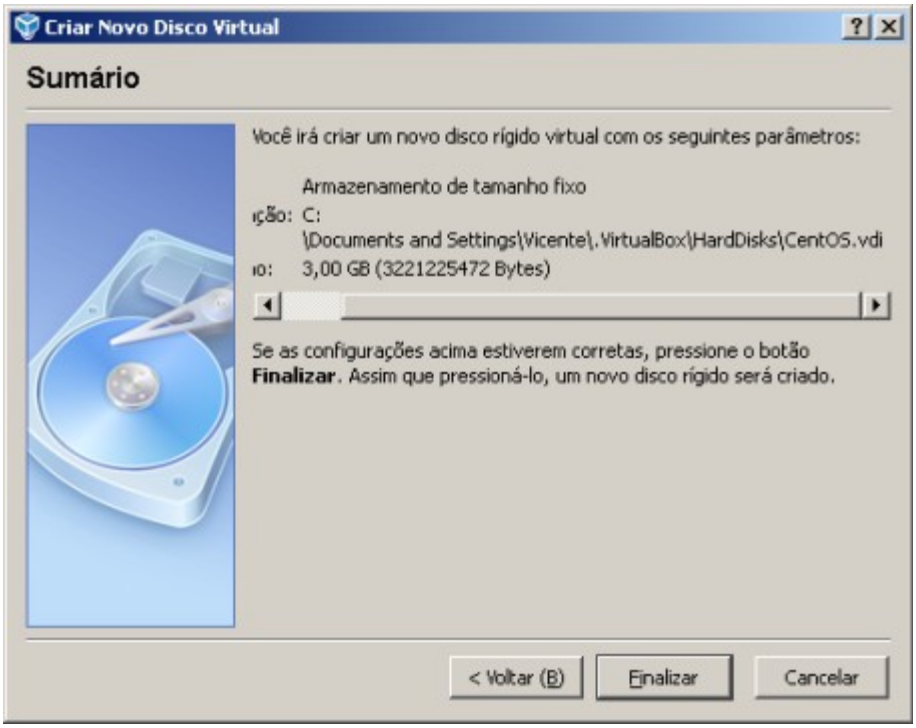
Será aberto um outro assistente para criar um disco virtual e clique em **Próximo** para continuar:



Existem 2 opções para criar o seu disco: dinamicamente expansível e tamanho fixo. O primeiro permite que você crie um disco com um tamanho de 10 GB por exemplo, sem alocar imediatamente o espaço. A medida que o espaço for sendo usado, o espaço será alocado. O problema desta abordagem é que o arquivo pode ficar altamente fragmentado, degradando a performance. O segundo tipo é o tamanho fixo. O espaço é alocado no momento da criação. Tende a ter melhor desempenho que o de tamanho dinamicamente expansível. Nesta condição escolha o **tamanho fixo** e clique em **Próximo**:

Nomeie o disco, escolha o tamanho e clique em **Próximo**:

Clique em **Finalizar** para continuar a criação do disco:

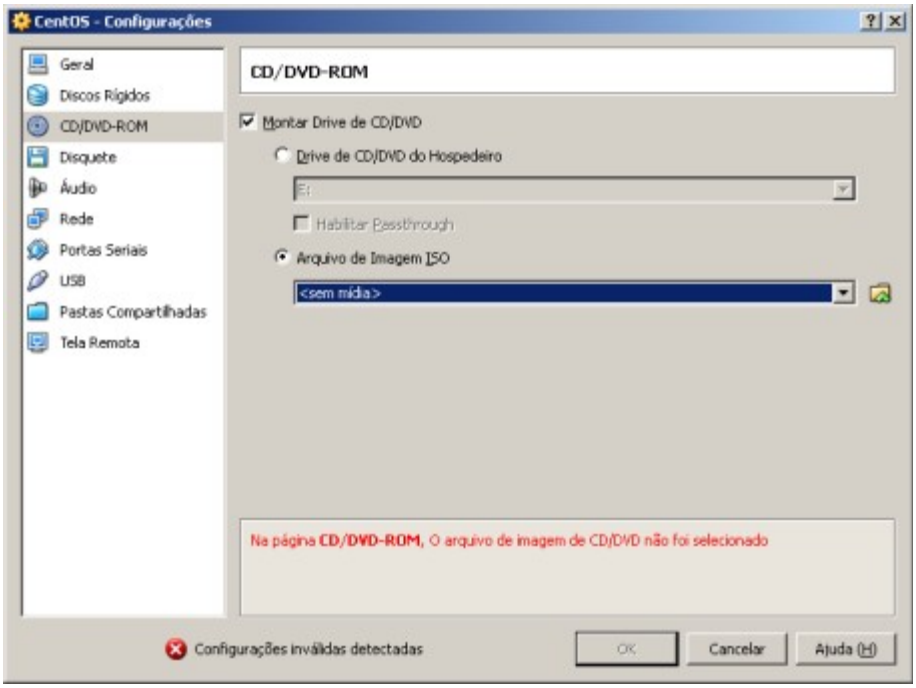


Aguarde a criação do disco:

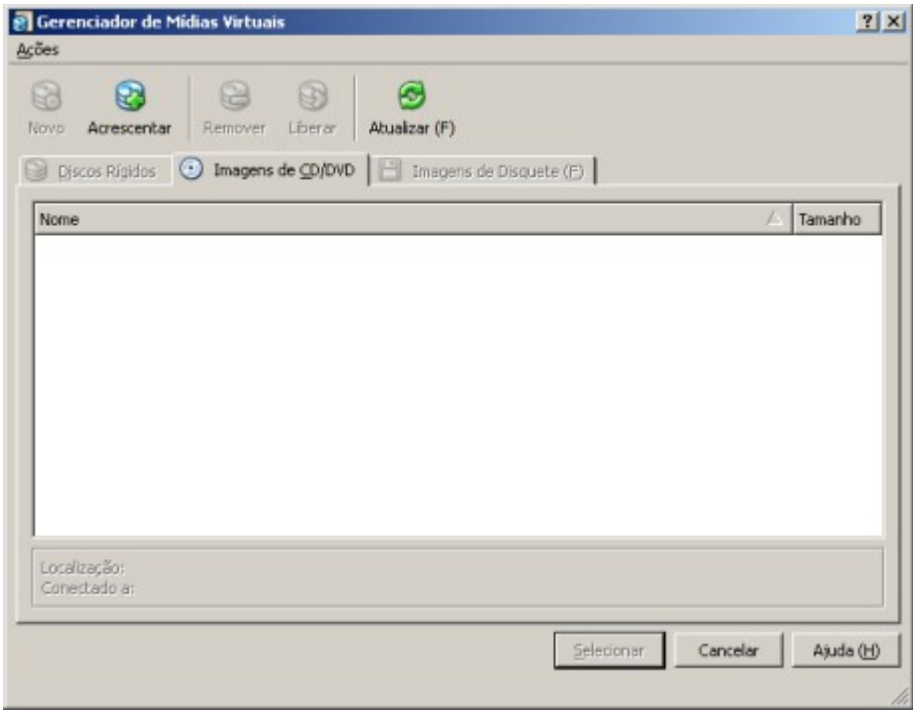
Depois que o disco foi criado, selecione-o e clique em **Próximo**:

Clique em **Finalizar** para terminar o assistente:

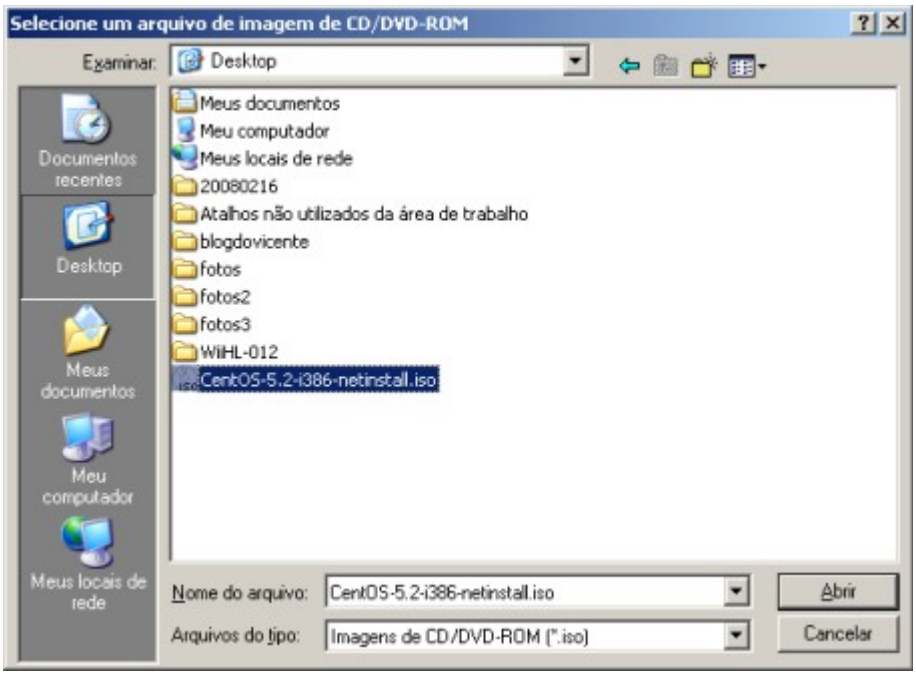
Sua máquina virtual está criada. No painel do lado esquerdo aparecerá o nome da máquina. No painel do lado direito, clique sobre o CD/DVD-ROM. Na tela que for aberta, selecione a opção **Montar drive de CD/DVD** e escolha se você usará o disco do seu drive de CD ou se você vai usar um ISO. Neste tutorial montaremos o ISO do CentOS. Clique na opção **Arquivo de Imagem ISO** e clique no botão em forma de Pasta:



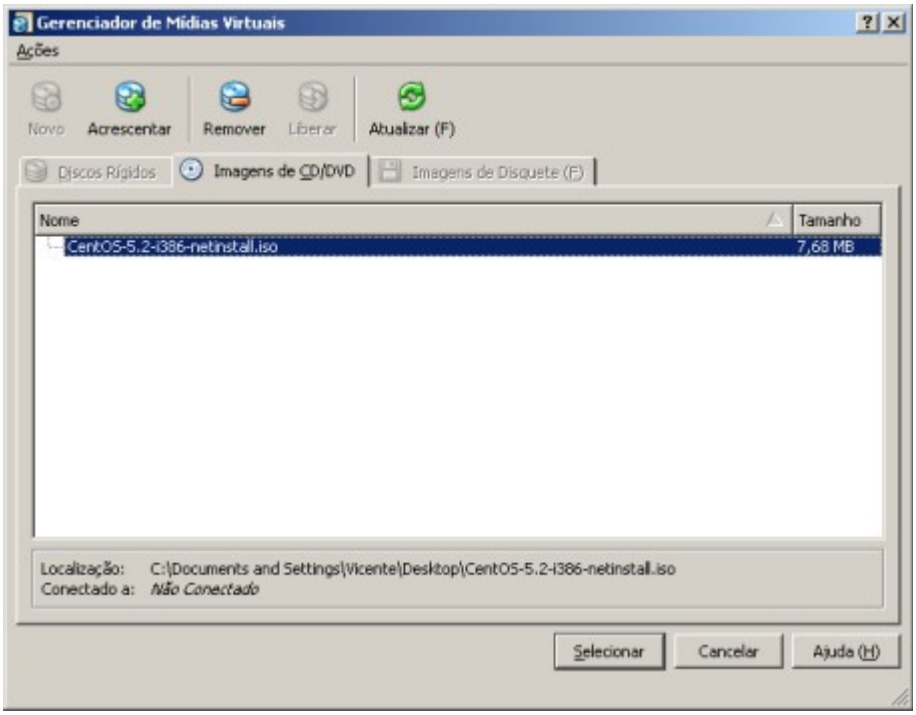
Será aberto um assistente de Mídias Virtuais. Clique em **Acrescentar** para adicionar um ISO:



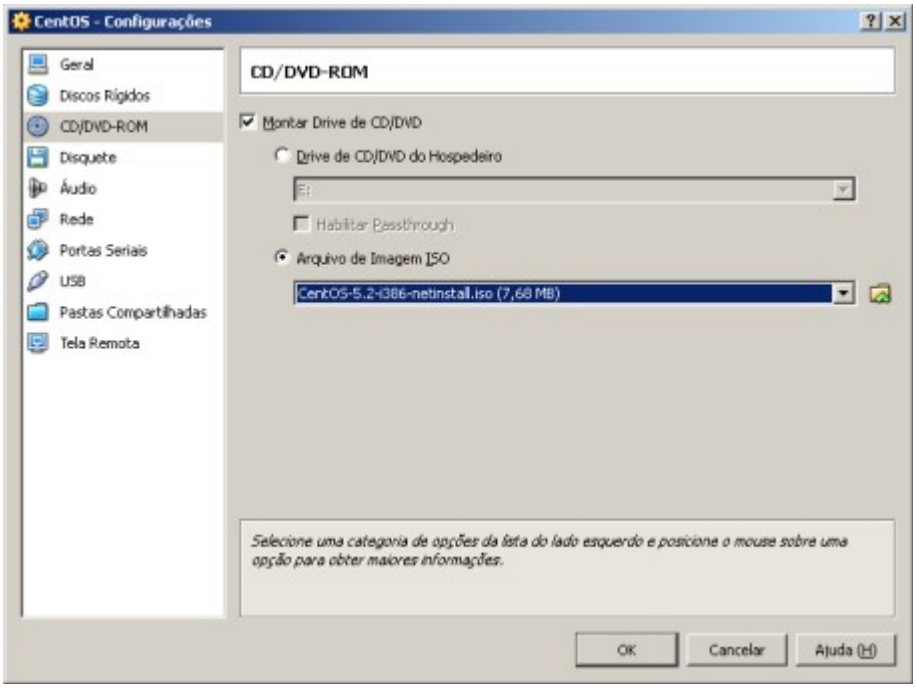
Selecione o arquivo de imagem e clique em **Abrir**:



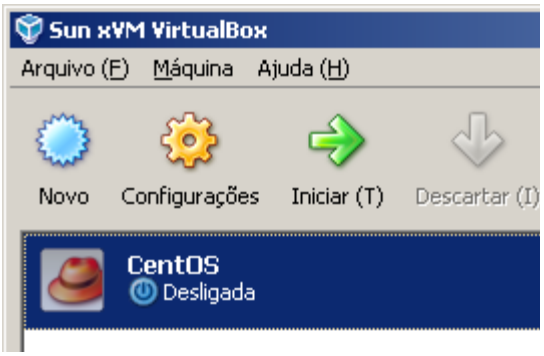
Clique no ISO que você acabou de adicionar e clique em **Selecionar**:



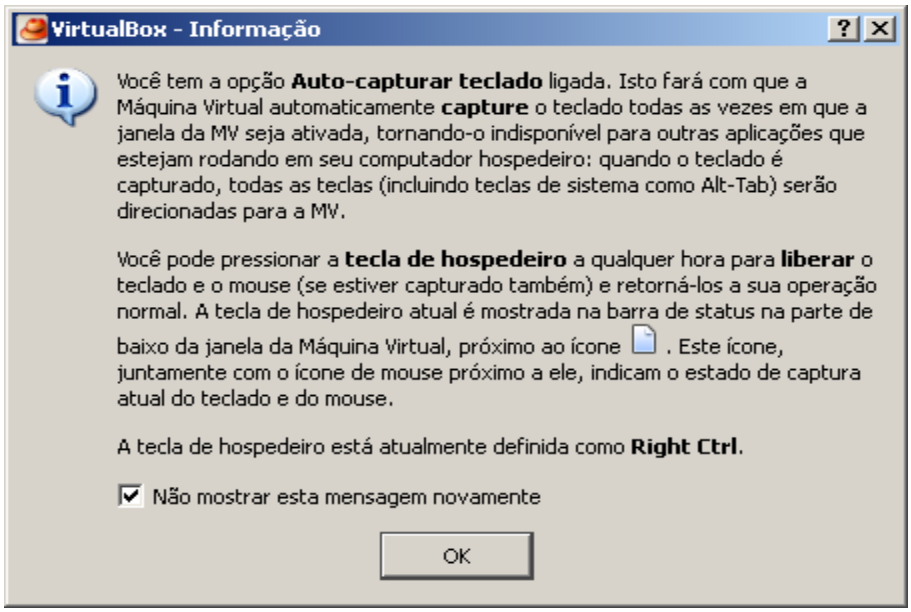
Clique em **OK** para finalizar a montagem do ISO na máquina virtual:

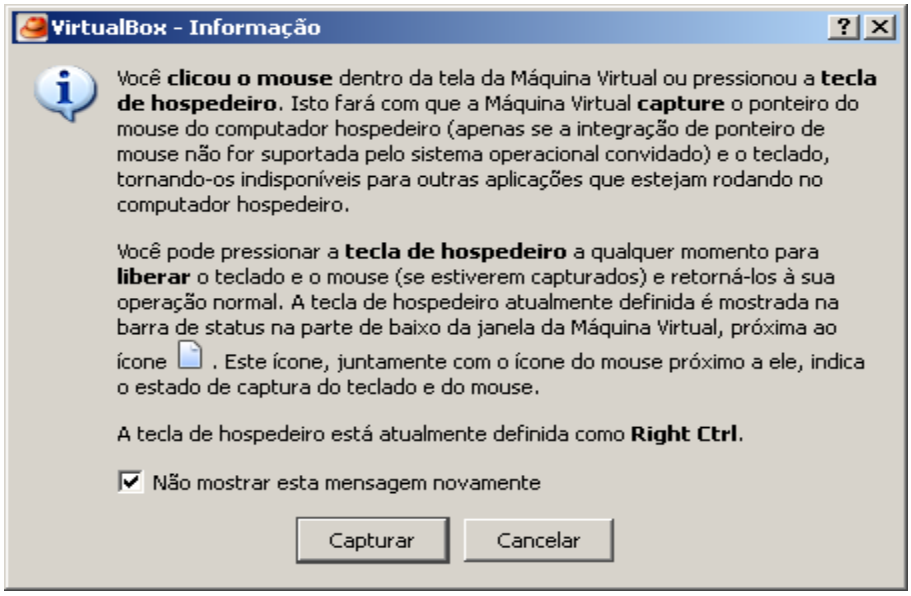


Agora você já pode iniciar a sua máquina virtual selecionando-a e clicando no botão **Iniciar**:

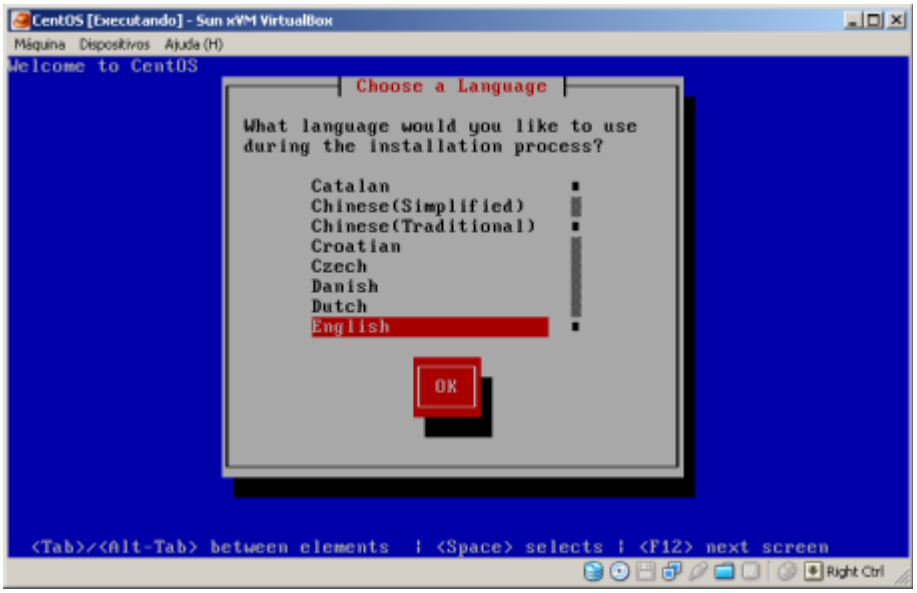


Quando você clicar na console da máquina virtual, você receberá alguns avisos. Eles informam que se você precisar sair da console, deverá pressionar a tecla **Ctrl do lado direito**. Para não receber mais estes avisos marque a opção **Não mostrar esta mensagem novamente**:





Agora você pode usar sua máquina virtual exatamente como se ela fosse uma máquina física:

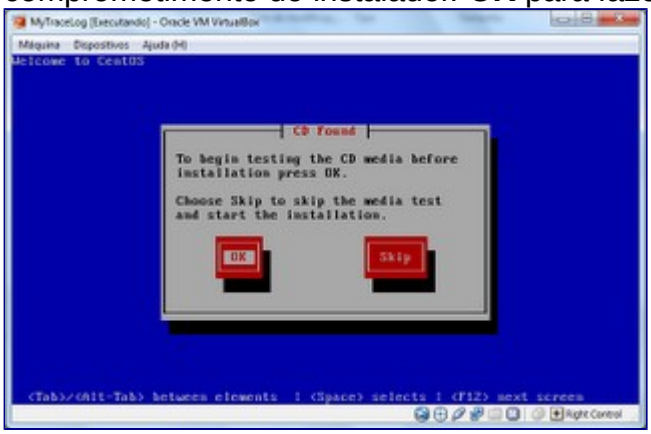


Instalando o CentOS 5.6

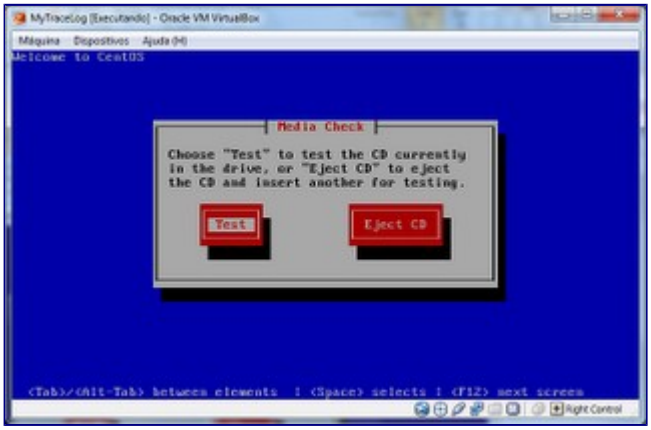
Automaticamente a VM irá bootar pelo DVD e iniciar a instalação. Dê enter para.

Selecionar a instalação em modo texto

1. Nesta tela é apresentado o teste de mídia, para garantir que não houve comprometimento do instalador. **OK** para fazer o teste e ***skip*** para pular essa etapa..

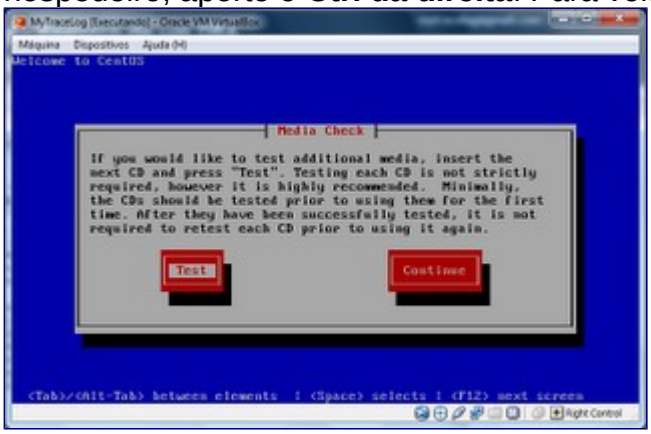


2. Dê **enter** em **OK**

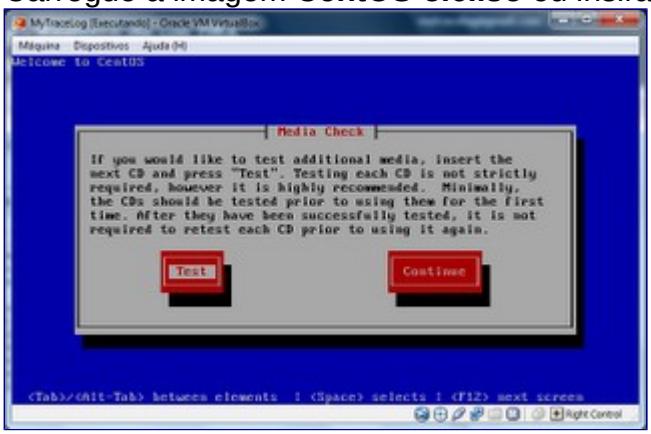


3. **Dê enter em Test para iniciar o teste.**

4. Carregue a imagem **CentOS-5.6.iso** ou insira o segundo DVD, então dê **enter** em **Test**. Detalhe caso for carregar a imagem, para assumir o controle do sistema hospedeiro, aperte o **Ctrl da direita**. Para voltar pra VM click dentro da janela.

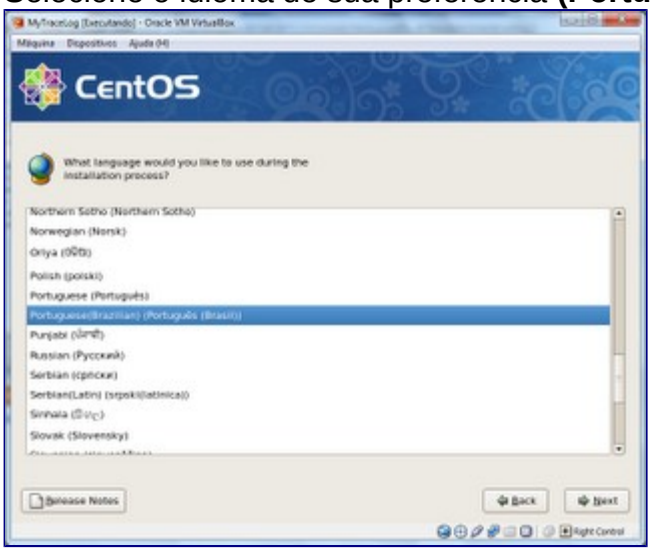


5. Carregue a imagem **CentOS-5.6.iso** ou insira o primeiro DVD.

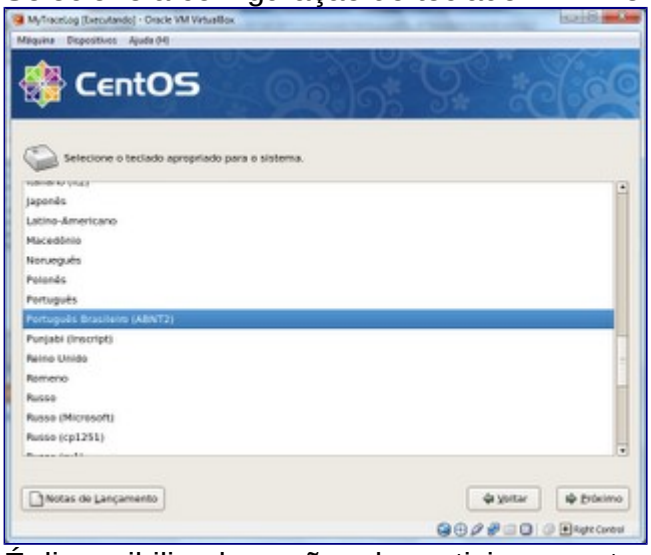


6. Finalmente iniciado o modo gráfico, click em **Next**.

7. Selecione o idioma de sua preferencia **(Português(Brasil))**

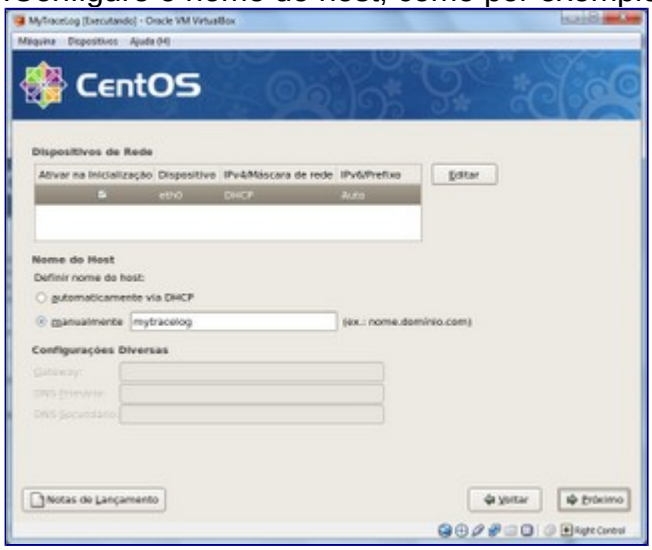


8. Selecione a configuração do teclado. Ex.: **Português Brasileiro (ABNT2). Próximo**.



9. É disponibilizado opções de particionamento do disco. Click em **Próximo**.

10. Configure o nome do host, como por exemplo demetrio.



11. Defina a sua região

.

12. Escolha uma senha para o usuário root, por exemplo demetrio.

13. Mantenha o opção Desktop - Gnome, click em **Próximo**.

14. Última tela antes de iniciar a instalação, click em **Próximo**.

15. Instalação concluída! Desmonte a imagem ou remova o DVD do drive. **Reinicializar**.

Reiniciado a máquina, precisamos fazer a configuração:

1. Vamos começar a configurar, click em **Avançar**.

2. Desabilite o Firewall. **Não use isso como prática em um sistema de produção**.

3. Desabilite o SELinux. **Não use isso como prática em um sistema de produção**.

4. Crie um usuário, como por exemplo usuário **mytracelog** e senha **mytracelog**.

5. Efetue o teste de som, click em **Avançar**.

6. Tela pra instalação de CDs adicionais, click em **Terminar**.

SHELL SCRIPT – Básico

Desenvolver...

Conclusão

Espero que tenha gostado. :)

Dúvida ou sugestões: chds_@hotmai.com