CEDUP – Centro de Educação Profissional Diomício Freitas de Tubarão

CURSO : Técnico em Redes de Computadores

Professor : Fernando Antônio Mattar Costa

Elaboradores : Maurício da Costa Canto e Erick Cavalcante

SISTEMAS OPERACIONAIS I

Tubarão Fevereiro de 2013

SUMÁRIO

1. SISTEMAS OPERACIONAIS

Um sistema operacional pode ser resumido como um conjunto de rotinas a ser executada pelo processador, a exemplo dos outros programas, porém servindo de gerenciador de recursos. O objetivo é de ser uma interface entre usuário e máquina, organizando e controlando hardware e software.

Todos os equipamentos eletrônicos que possuem algumas funcionalidades mais avançadas possuem um S.O. (Sistema Operacional) rodando neles. Televisores, aparelhos de som e até mesmo os eletrodomésticos já possuem S.O's. E todos estes S.O's são uma evolução natural dos sistemas feitos para gerenciar os computadores. Sem um sistema desse tipo, o computador se torna inútil nos dias atuais.

Definições importantes relacionadas a SO são:

Kernel – O núcleo do sistema, faz os gerenciamentos de memória e processos, periféricos e dispositivos (baixo nível).

Shell – Linguagem de programação que 'traduz' os comandos do usuário para o kernel ou o contrário.

Script – São conjuntos de comandos, que normalmente seriam executados no prompt isolados, sequenciais com diversas possibilidades de condicionais.

Distribuições – É o kernel modificado acompanhado por uma série de aplicações e ferramentas formando um SO. Cada uma pode oferecer diferente tipos de aplicativos.

1.1. WINDOWS

Existem diversos tipos de S.O's. Os mais conhecidos são os da família Windows da Microsoft. Que segue duas linhas de desenvolvimento. A primeira, que é a que convivemos praticamente todos os dias, destinada para usuários domésticos e uso diário. Esta linha tem os produtos Windows XP, Vista e Seven como os mais conhecidos atualmente. Já a segunda linha é da de S.O's para servidores. Que possui os Sistemas Windows Server 2003, 2008 e o recém-lançado 2012. Essa linha possui uma quantidade muito maior de recursos avançados.

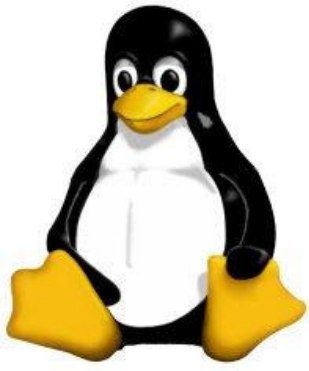
Muitas empresas possuem pelo menos um servidor de seu datacenter com S.O. da família Windows Server. Isso porque o parque de computadores das empresas costuma ser baseada na outra família de S.O. da Microsoft. Isso torna necessário pelo menos um conhecimento básico sobre este S.O.

1.1.1 SOFTWARE LIVRE X PROPRIETÁRIO

Entende-se por software livre todo programa computacional que atende as liberdades de estudar, aperfeiçoar, executar e redistribuir os mesmos. O software proprietário por sua vez veta o código-fonte para modificações, inviabilizando alterações.

1.1.2 LINUX

O célebre linux foi desenvolvido pelo então estudante finlandês Linus Torvalds, é um dos estandartes do Sofware Livre. Utiliza a política GPL (General Public License), licença pública geral, que permite a alteração de seu código fonte por qualquer usuário.



1.2 UNIX

Criado nos anos 60 por Kenneth Thompson é considerado o pai dos SO's, por inovar na época de sua criação com rotinas usadas ainda hoje por diversos SO's. O próprio Linux é derivado do Unix. Dentre os derivados também temos a família BSD, abordaremos adiante um SO dessa família o FreeBSD.

1.2.1. História do FreeBSD

O FreeBSD surgiu a partir do 386BSD , mais exatamente de um patchkit (pacote de correção de erros) do 386BSD. Na época, o 386BSD era considerado um bom BSD, mas que estava enfrentando alguns problemas graves na questão de manutenção das atualizações do sistema.

Então, um time formado pelos responsáveis pelo desenvolvimento deste patchkit (Nate Williams, Rod Grimes e Jordan Hubbard) resolveram levá-lo ao conhecimento do mantenedor do projeto, Bill Jolitz , para que com estes pacotes fosse possível atualizar o sistema de forma bastante prática.

Mas para a surpresa do trio, Bill Jolitz não viu com bons olhos as intenções dos desenvolvedores, retirando apoio ao projeto bem como nenhum planejamento futuro para o desenvolvimento do sistema.

Todos estes acontecimentos dataram pelos anos de 1992 e 1993, sendo que neste último ano realmente podemos considerar que o FreeBSD foi concebido pelos antigos desenvolvedores do patchkit do 386BSD.

Sendo assim o FreeBSD (nome sugerido por David Greenman), nasce para continuar um trabalho de base já realizado no 386BSD. E como uma das metas do projeto era justamente divulgar um pouco mais o sistema que na época era praticamente desconhecido, resolveram entrar em negociações Walnut Creek CDROM.

Na época poucas pessoas possuíam conexões com a Internet e até pelo fato da grande rede estar engatinhando, a grande jogada era realmente distribuir o FreeBSD em CD. De fato a Walnut aceitou a distribuir o CD com o FreeBSD, e fez algo a mais muito válido que certamente ajudou o FreeBSD a ser o que é hoje, que foi o fato de disponibilizar uma máquina dedicada e uma boa conexão com a Internet.

Realmente o feito realizado pela Walnut abriu as portas para o FreeBSD, pois na época a Internet era muito usada pelos meios acadêmicos, possibilitando a rápida propagação do conhecimento do projeto do FreeBSD.

O FreeBSD foi lançado oficialmente em CD e na Internet em dezembro de 1993, baseado no 4.3BSD-Lite, conhecido como Net/2. Na época a versão denominava-se FreeBSD 1.0, tendo como base muitos componentes do 386BSD e da Free Software Foundation.

Segundo os autores do projeto o sucesso foi bastante empolgante, fazendo com que em maio de 1994 surgisse o FreeBSD 1.1. Mas, nem tudo é conto de fadas no desenvolvimento do FreeBSD. Na época, existia um processo judicial envolvendo a Novell e a U.C Berlekey sobre a fita do Net/2.Muitos pedaços do código fonte do Net/2 que na época eram a grande base do FreeBSD , tinham como proprietária a AT&T, esta comprada pela Novell.

Sendo assim, a solução encontrada foi retirar todos os trechos de propriedade da Novell do sistema e refazer todos os setores do sistema que ficaram órfãos . Mas mesmo assim, ainda foi lançada uma versão do FreeBSD, a 1.1.5.1.

Ainda neste tempo foi lançado o 4.4BSD-Lite, sendo um BSD com todos os códigos envolvidos em decisões legais, retirados do sistema. Este fato deixou este BSD

muito incompleto para tornar-se realmente utilizável. E nesta atmosfera de códigos incompletos, é que o FreeBSD ganha força, tendo grande parte do sistema renascida, pois esta é a melhor forma de demonstrar a difícil situação desta época.

Já em dezembro de 1994, sai a versão 2.0 do FreeBSD. Como de costume sua principal fonte de distribuição foi o CD, mas a Internet também ajudou muito a disseminar a nova versão do sistema. Caiu nas graças do público mesmo com seus incontáveis problemas, e em Junho de 1995 sai a versão 2.0.5.

Segundo os desenvolvedores do projeto, a partir de agosto de 1996 o FreeBSD 2.1.5 passa a ser utilizado em provedores de Internet e no ramo corporativo. A partir daí o seu desenvolvimento começa a tornar-se mais organizado, surgindo o ramo 2.1-STABLE.

Em Fevereiro de 1997 a série 2.1-STABLE tornou-se completa, abrindo caminhos para o desenvolvimento do ramo 2.2. Mas um pouco antes, em novembro de 1996 já eram iniciados os trabalhos para o desenvolvimento desta nova série. Muitas pessoas ainda lembram-se desta data, pois foi deste ponto que o FreeBSD adquiriu denominações para as fases de desenvolvimento, tais como RELENG e CURRENT.

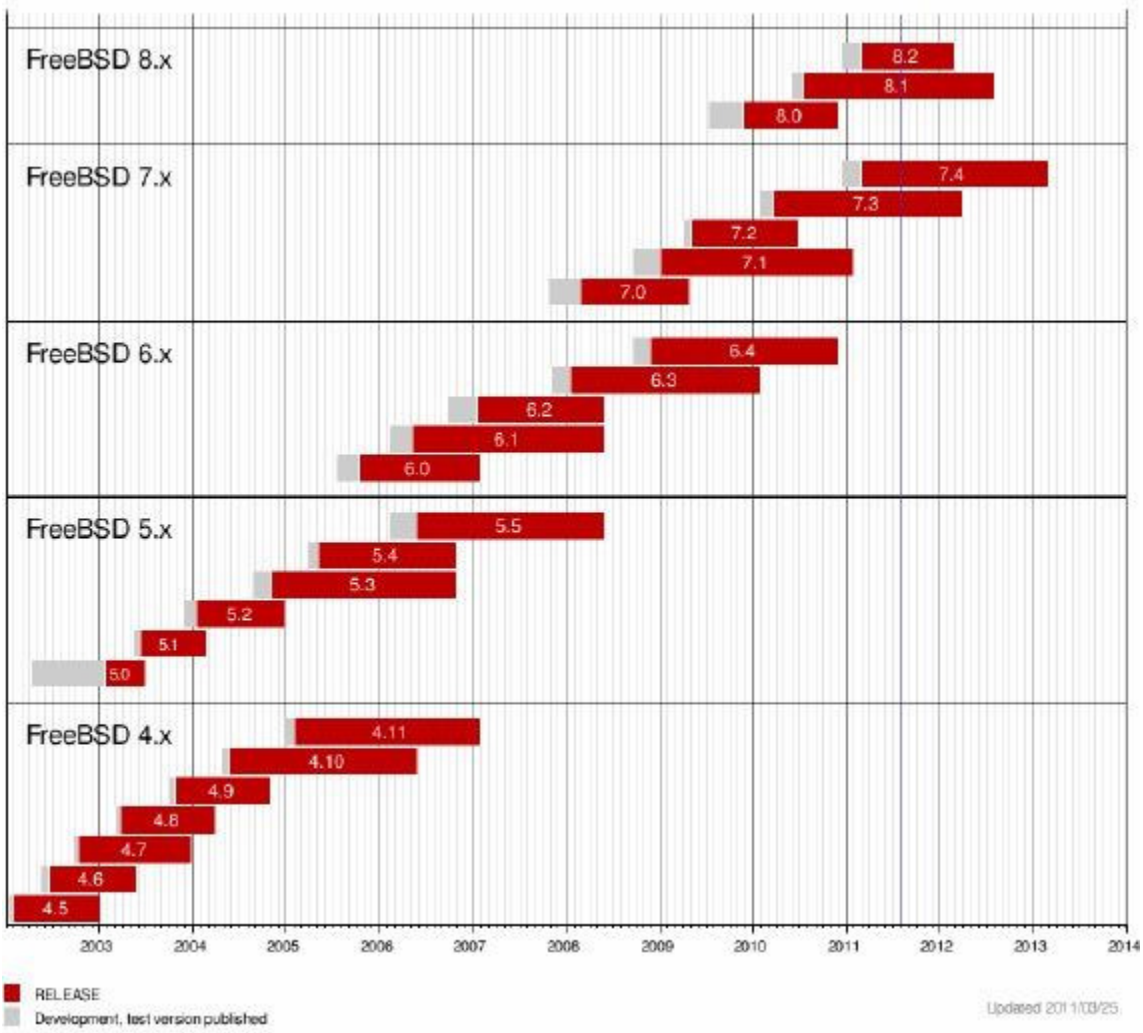
Da série RELENG_2_2, surgiu em Abril de 1997 o FreBSD 2.2.1 e a última em Novembro de 1998. Surgiram então duas fases de desenvolvimento a partir desta época. A série 3.X STABLE, e a 4.0-CURRENT. Depois de vários aperfeiçoamentos, a série 3.X do sistema tornou-se completo em Junho de 2000.

Um pouco antes, em Março de 2000 já estava em ambiente STABLE a série 4.X, que é por muitos considerada a melhor versão do sistema para uso em ambientes de produção . Esta versão contou com muitos adicionais de suporte a hardware e melhoria em performance do sistema em relação à série 3.X, ganhando finalmente a confiabilidade tanto tempo desejada. Esta confiabilidade trouxe muitos adeptos ao sistema, surgindo um novo leque de usuários, que são os ex-usuários de Linux. Este ex-usuários conhecendo o FreeBSD de forma mais profunda, admiram-se pela qualidade do projeto do FreeBSD , ocorrendo um gradual aumento do uso do FreeBSD no seu dia-a-dia.

A versão 5.0-RELEASE chega ao público em geral em 19 de janeiro de 2003. Nesta série, há um suporte muito maior a novas tecnologias computacionais, tais como:

- Suporte avançado a multiprocessamento simétrico, suporte avançado a aplicações.
- Multithread e suporte a novas arquiteturas como UltraSparc® e ia64.

Desta etapa surge alguns acontecimentos muito curiosos, tais como a divisão de alguns usuários em conservadores, por utilizarem a versão 4.X (mais estável) e os mais audaciosos utilizando a série 5.X, que por esta época apesar das novidades era altamente instável para ser utilizada em ambiente de produção.



1.2.2. Por que o FreeBSD tem a sua fama?

Por que um sistema operacional aparentemente complexo de usar, com uma publicidade de uso totalmente diferente do Linux consegue fazer tanto sucesso com os profissionais de informática, tais como usuários avançados e administradores de rede?

Para começo de conversa, o FreeBSD conta com recursos de instalação de softwares e atualização do sistema muito superior ao Linux. Existe um repositório (árvore de aplicações) para o FreeBSD conhecida como ports, ou seja, softwares portados para o FreeBSD , inclusive aplicações Linux. Lembrando que o FreeBSD possui a vantagem de emular compatibilidade Linux. A vantagem real do ports é simplesmente baixar o código-fonte do software, compila-lo e instalar no sistema . Desta forma para cada hardware onde há o sistema rodando há uma otimização real da performance, diferentemente do Linux com o seu apt-get que simplesmente pega o binário e joga no sistema, sem otimização nenhuma.

Outro ponto importante é a organização do desenvolvimento do sistema. A equipe do FreeBSD conta com um time de commiters , que são os responsáveis por gravar as informações no repositório CVS do sistema. E antes de qualquer procedimento de atualização de código, seja este enviado por um usuário interessado em ajudar o desenvolvimento do sistema ou por um próprio commiter, há uma rígida análise no código, para evitar problemas futuros de estabilidade e segurança.

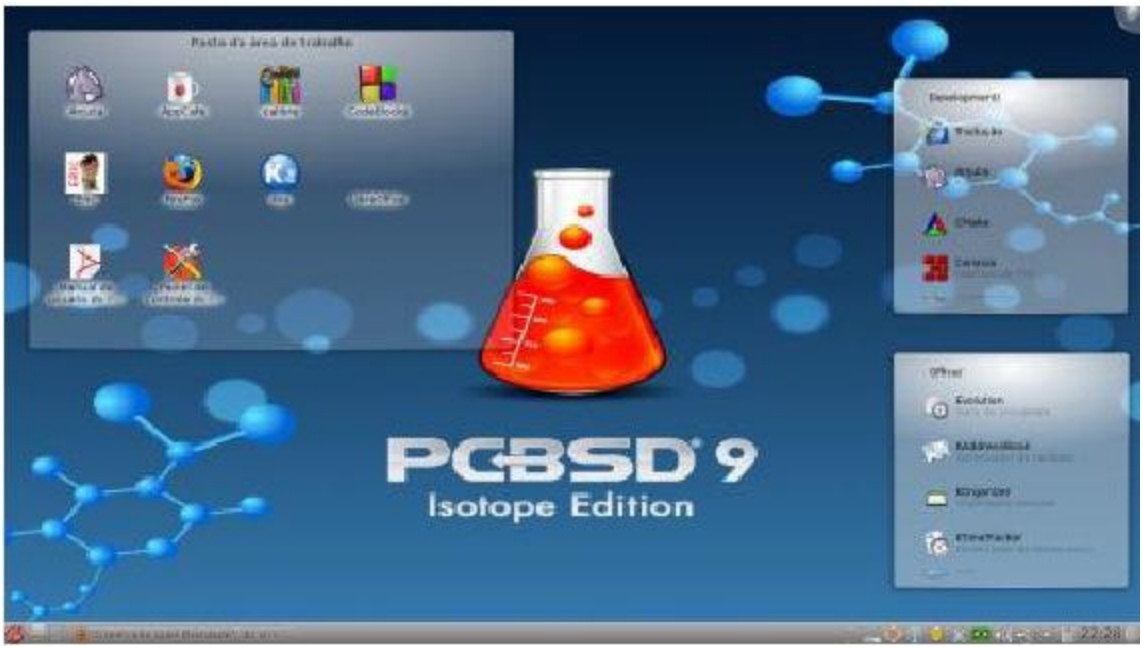
Esta construção robusta e aparentemente burocrática faz o FreeBSD ser muito seguro na execução de tarefas de alta disponibilidade. O sistema agüenta uma carga muito maior de processamento de tarefas que qualquer sistema da Família Linux. Não é a toa que muitas empresas e provedores o usam para execução de tarefas pesadas e importantes.

A fama do FreeBSD também é atribuída à organização do projeto no sentido de todos os diretórios de configuração, manuseio de parâmetros e a disposição inteligente de sua árvore de diretórios (/etc , /usr e /var) mudarem pouco com a atualização das versões.

Um sistema desenvolvido para rodar em FreeBSD 3.X certamente irá rodar no FreeBSD 5.X, ou um usuário que usava o FreeBSD 4.X irá utilizar sem problemas a série 5X. Esta concordância entre as versões é um fator muito positivo para o FreeBSD. E para os usuários que possuem um hardware pouco poderoso, o FreeBSD também possui pontos interessantes, já que o LINUX com o passar dos anos está tornando-se muito pesado , transformando-se em um sistema totalmente gráfico e sacrificando muito os computadores mais obsoletos.

Certamente o usuário com um pouco de conhecimento de FreeBSD irá conseguir instala-lo em um Pentium 100 Mhz com 16 MB de memória e um disco de 1GB. Basta escolher os pacotes certos na hora da instalação para concluir com sucesso a instalação do sistema neste tipo de hardware.

O único fator negativo certamente é certo atraso em relação ao desenvolvimento de um sistema desktop realmente funcional para o usuário final, pesar do esforço empenhado pelos sistemas PC-BSD e DesktopBSD, que contribuem e muito para o desenvolvimento de uma linha mais gráfica para o FreeBSD. Certamente com o tempo teremos novidades nesta área, inclusive a internacionalização do projeto como um todo, pois ainda há uma forte tendência de toda a documentação, menus de instalação e parte gráfica ainda serem escritos para usuários de conhecimentos abrangentes na língua inglesa.

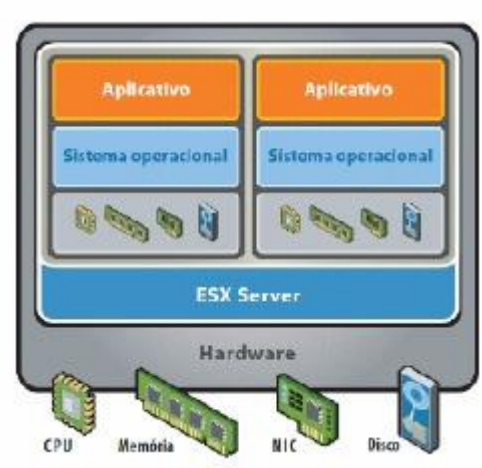


2. VIRTUALIZAÇÃO

É um conceito que vem ganhando cada vez mais força na área de tecnologia.

Ao falar em virtualização, é inevitável que a maioria das pessoas a associem à ideia de vários sistemas operacionais rodando na mesma máquina. Esse é, na verdade, um dos diversos tipos de virtualização: a de hardware. Se por um lado ela não é a única, por outro é, certamente, a mais perceptível.

Uma boa pesquisa no Google pode ser necessária para ampliar seus conhecimentos sobre virtualização, compartilhe suas dúvidas e aprendizado em sala .



2.1. VIRTUALBOX

VirtualBox é um software de virtualização desenvolvido pela empresa Innotek depois comprado pela Sun Microsystems que posteriormente foi comprada pela Oracle. Como o VMware Workstation, visa criar ambientes para instalação de sistemas distintos. Ele permite a instalação e utilização de um SO dentro de outro, assim como seus respectivos softwares, como dois ou mais computadores independentes, mas compartilhando fisicamente o mesmo hardware.

Essa ferramenta esta disponível para download no site www.virtualbox.org e é compatível tanto com S.O's da família Windows como os baseados em UNIX.

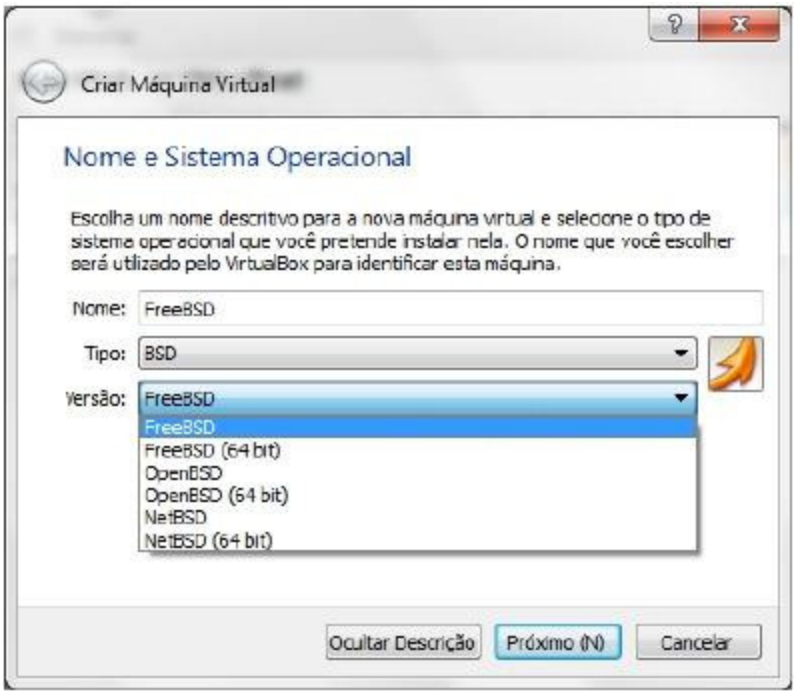
Acesse o link abaixo para mais detalhes sobre a instalação da VirtualBox. http://www.oracle.com/technetwork/pt/articles/virtualization/instalacao-oracle-virtualbox-4-2-513998-ptb.html

Geralmente a instalação de 'máquinas virtuais' (SO rodando sobre outro), é apoiada pelo uso de Imagens ISO (ou mídia de instalação física). Imagens ISO são gravações que contém todo os componentes de uma mídia, e sendo assim, possuem o tamanho real do CD ou DVD. A sigla significa International Organization for Standardization – Organização Interncional de Padronização. Todos os arquivos são mantidos sem compressão, e é uma extensão largamente usada para distribuição de SO's na internet.

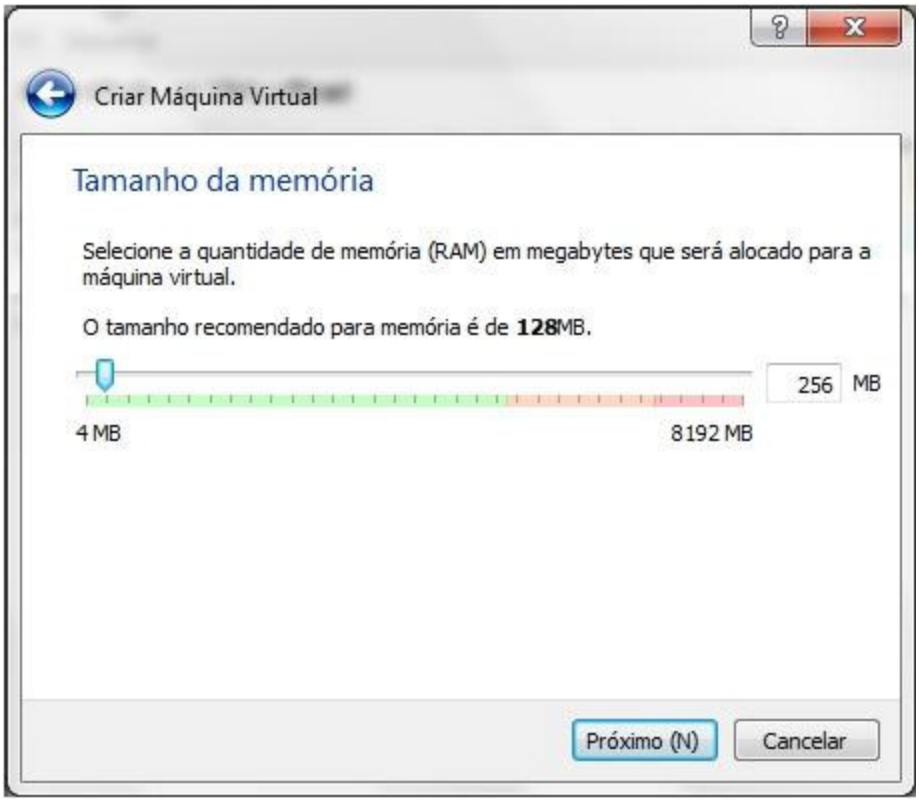
2.1.1. Criando Máquinas Virtuais

Supondo que você já seguiu os passos do link colocado no item anterior, vamos iniciar a criação de uma nova máquina virtual.

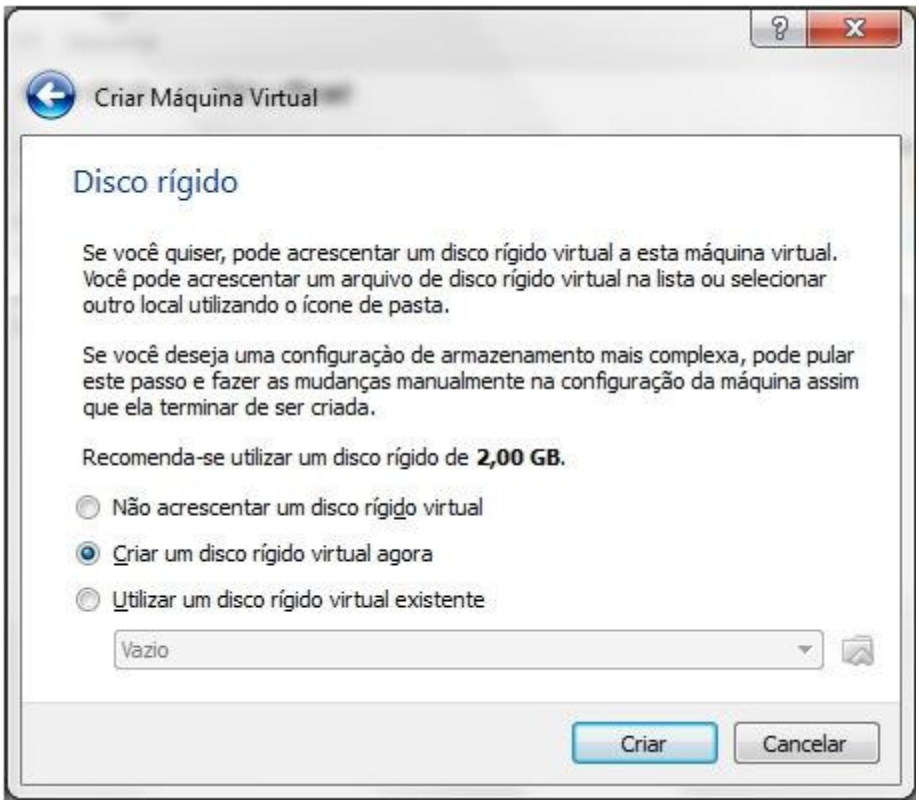
Tela inicial do VirtualBox. Para criar uma máquina nova é bem simples, basta iniciar clicando no botão Novo (em vermelho).



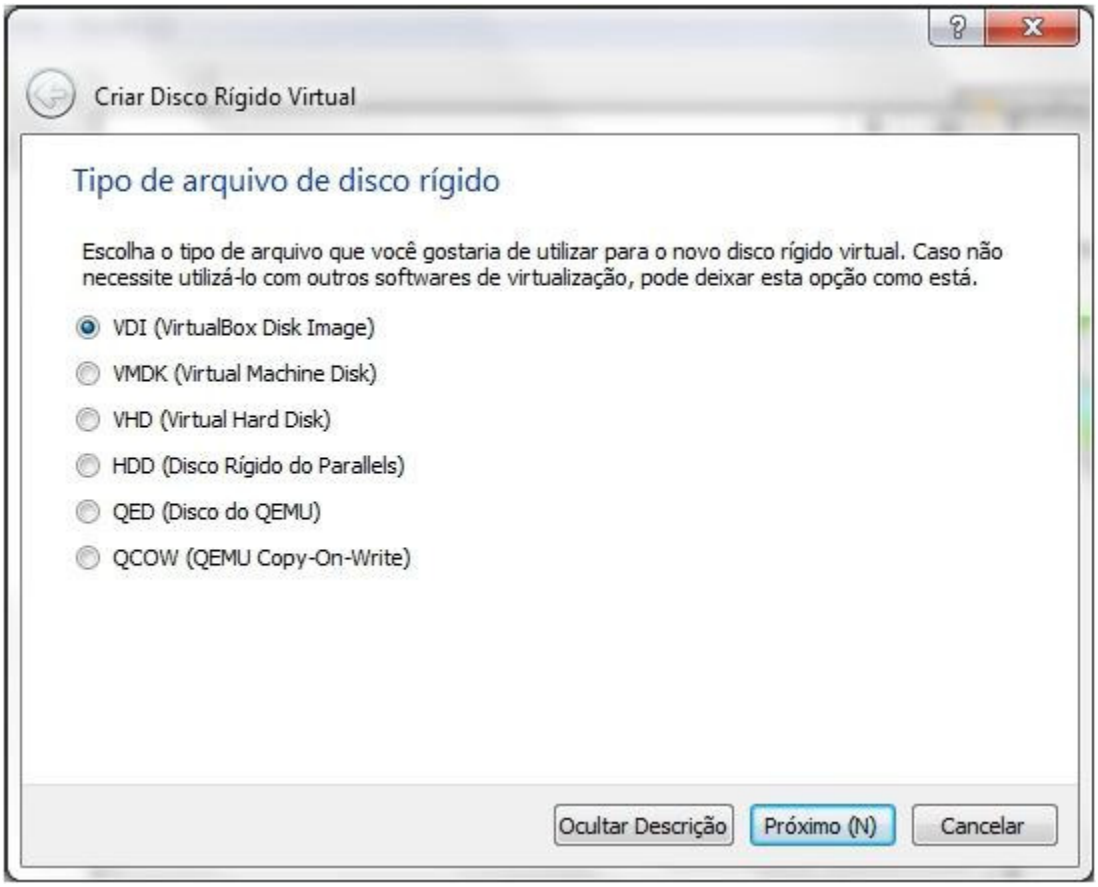
Nome da VM, tipo e versão do S.O que serão utilizados.



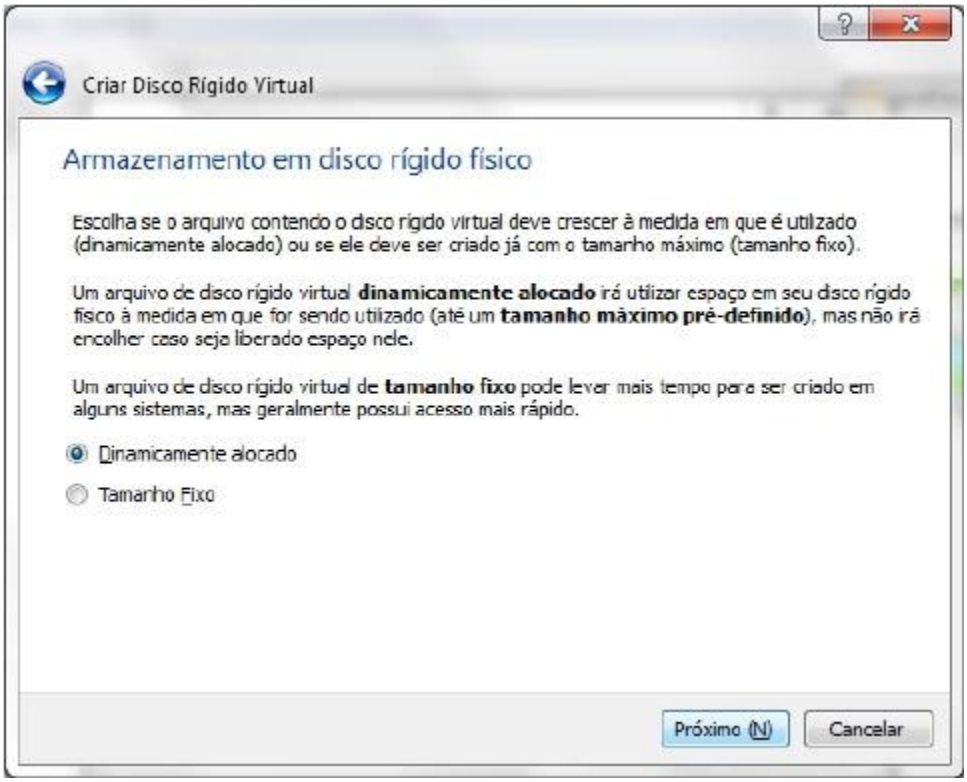
Quantidade de memória que será utilizada pela VM que está sendo criada.



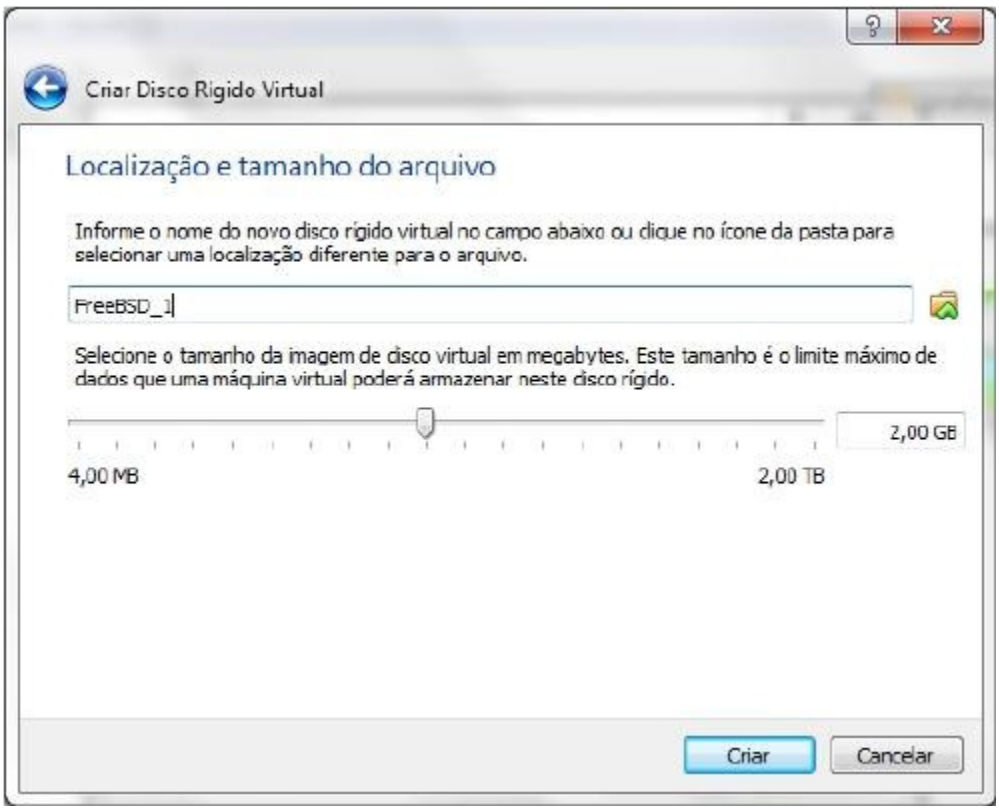
Disco rígido virtual que será utilizado pela VM.



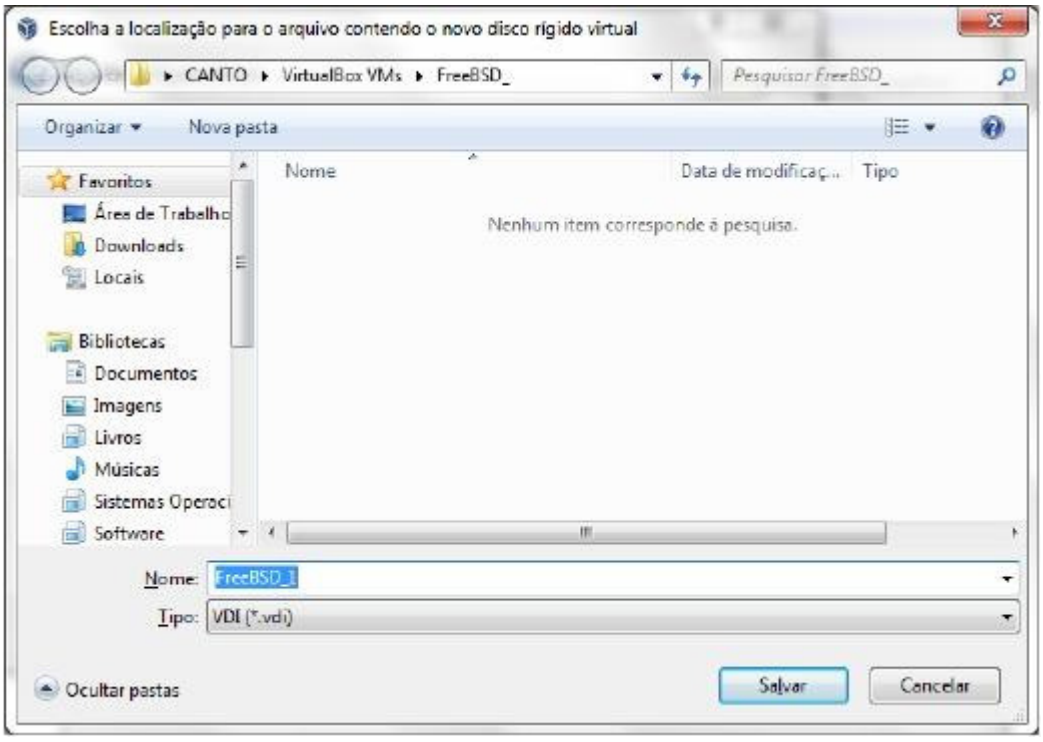
Tipo de disco utilizado pela VM.



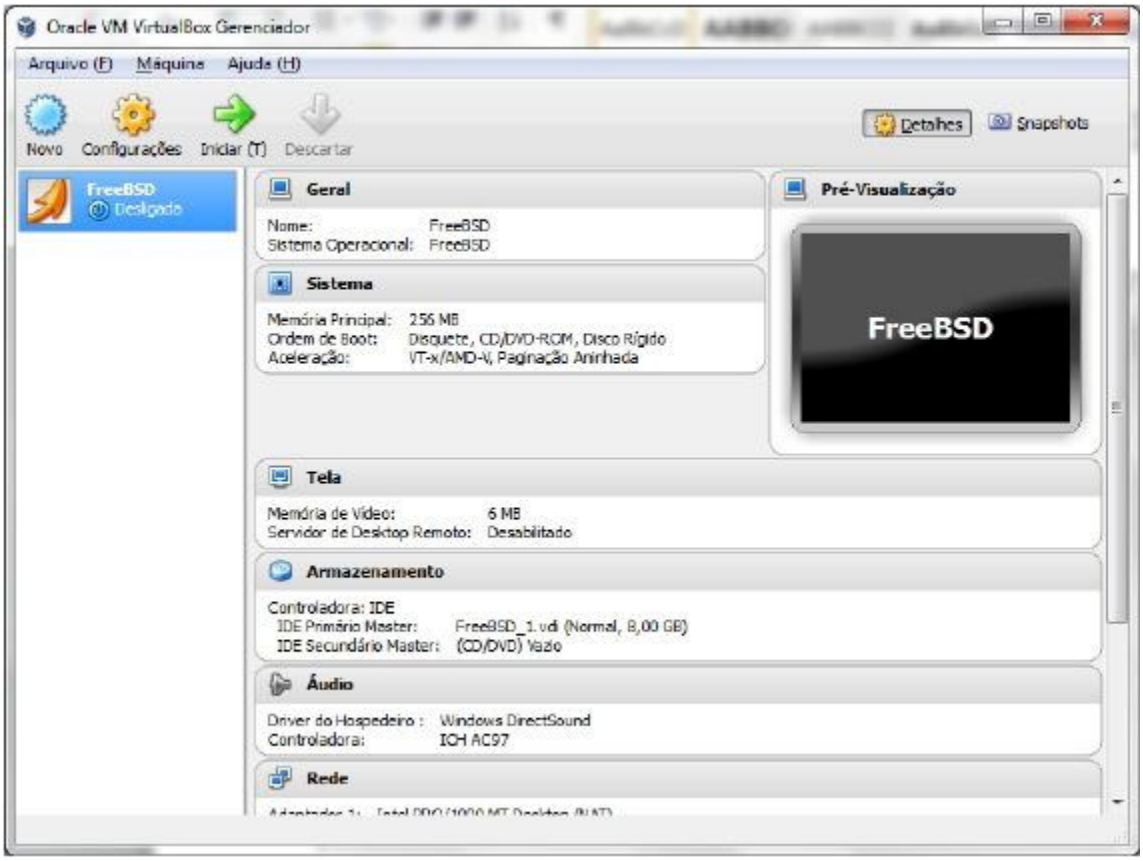
Tipo de armazenamento do disco da VM.



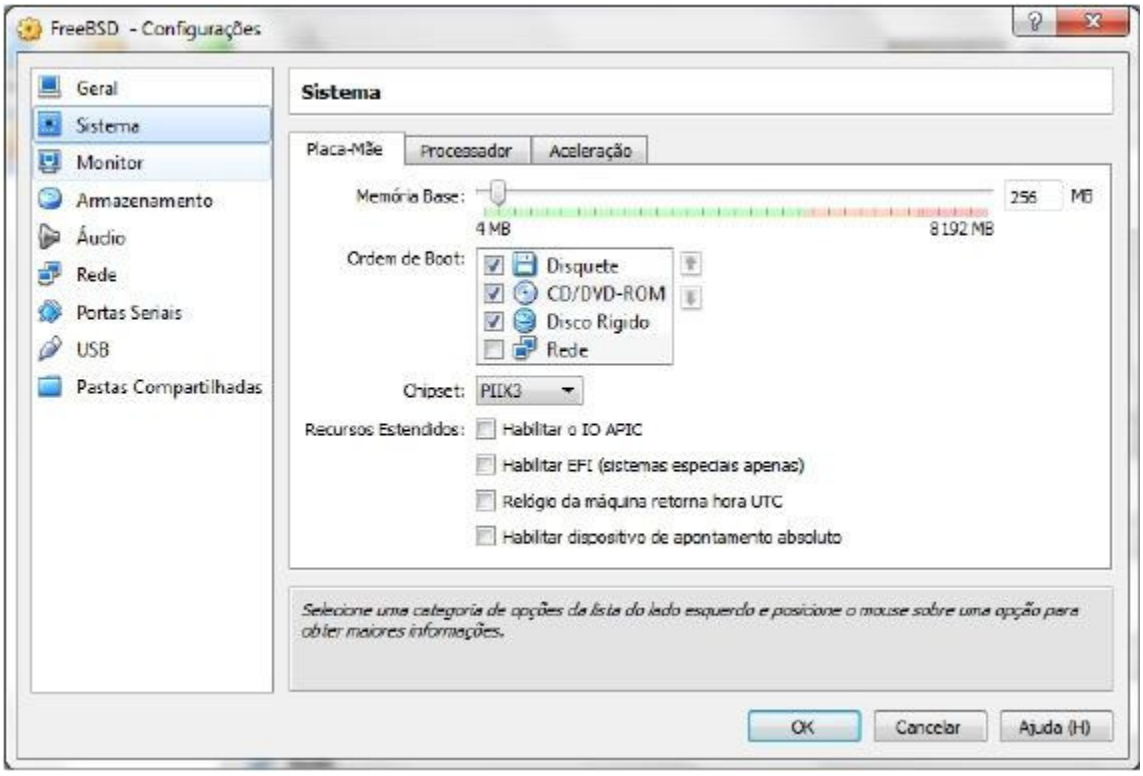
Local onde o disco virtual será salvo e o tamanho. Inicialmente podemos trabalhar com 8,00 GB



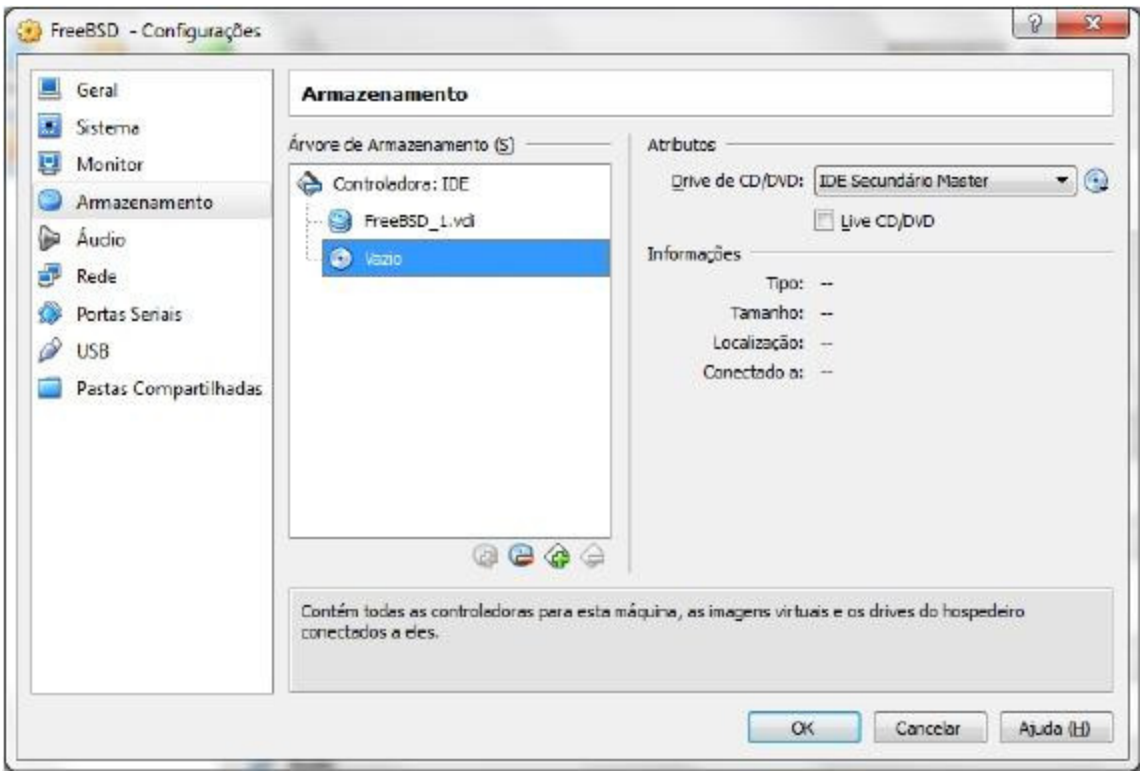
Local onde serão criados os discos.



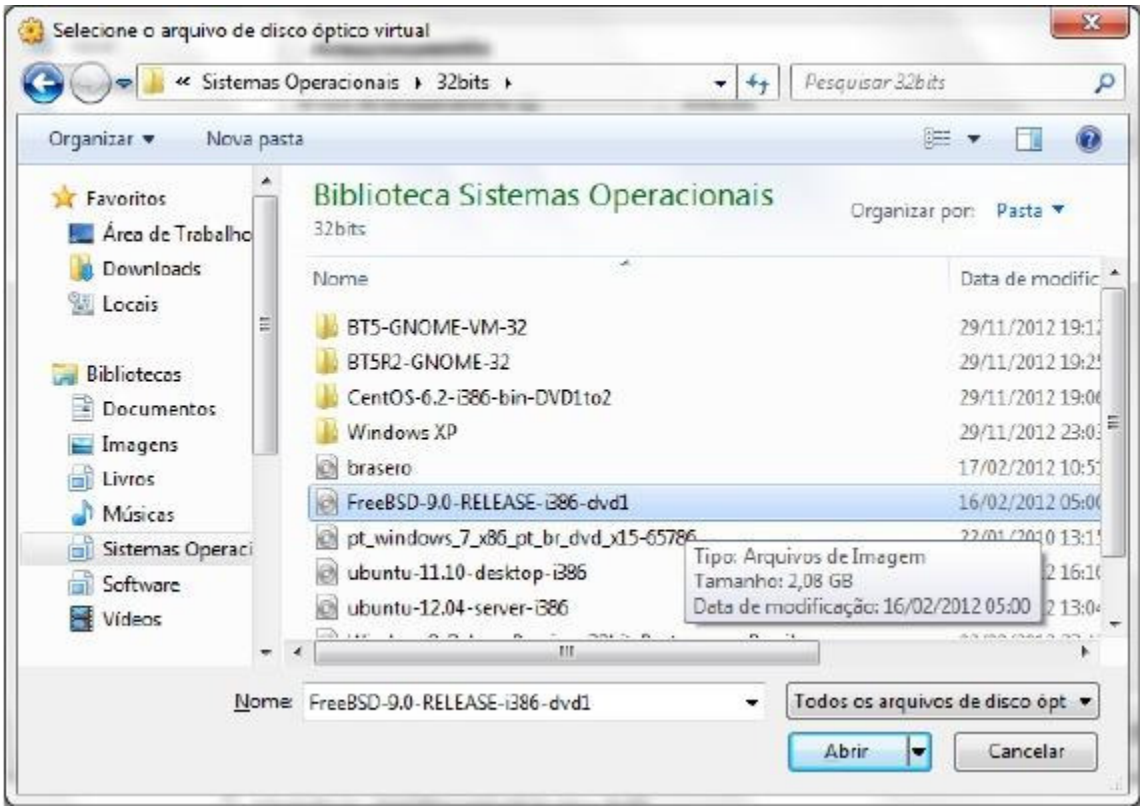
Pronto nossa máquina foi criada com sucesso.



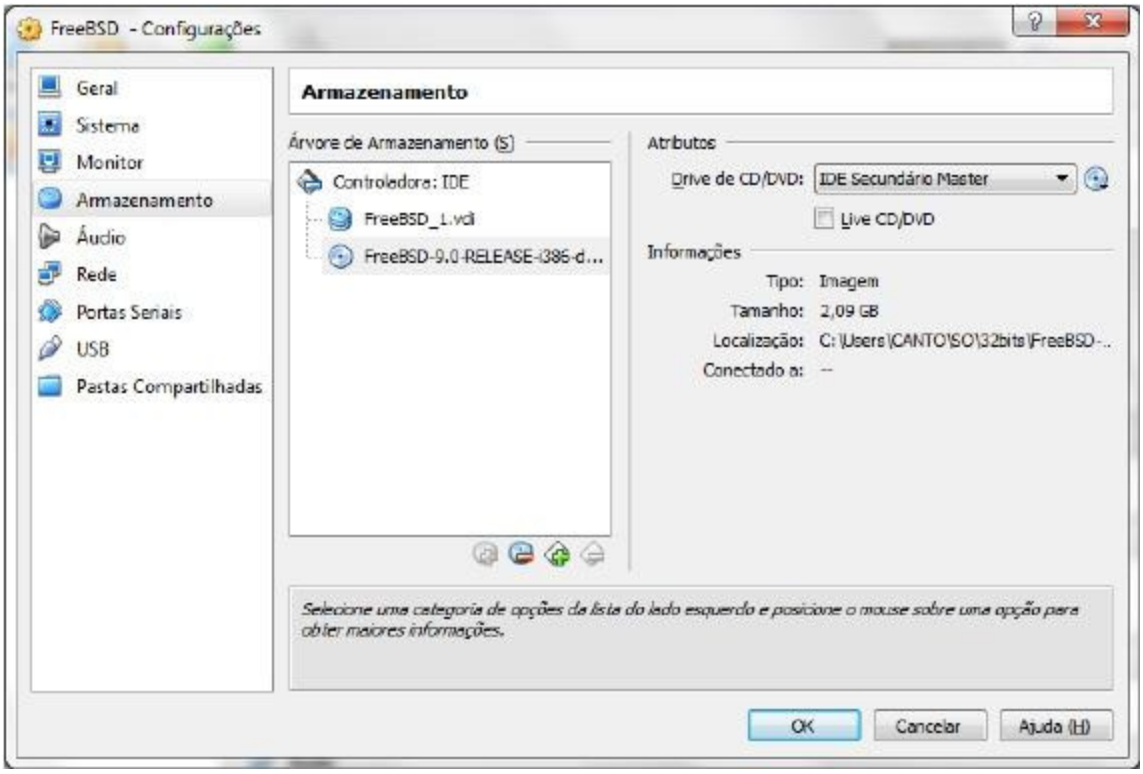
Esta é a tela das Configurações da VM que nós criamos.



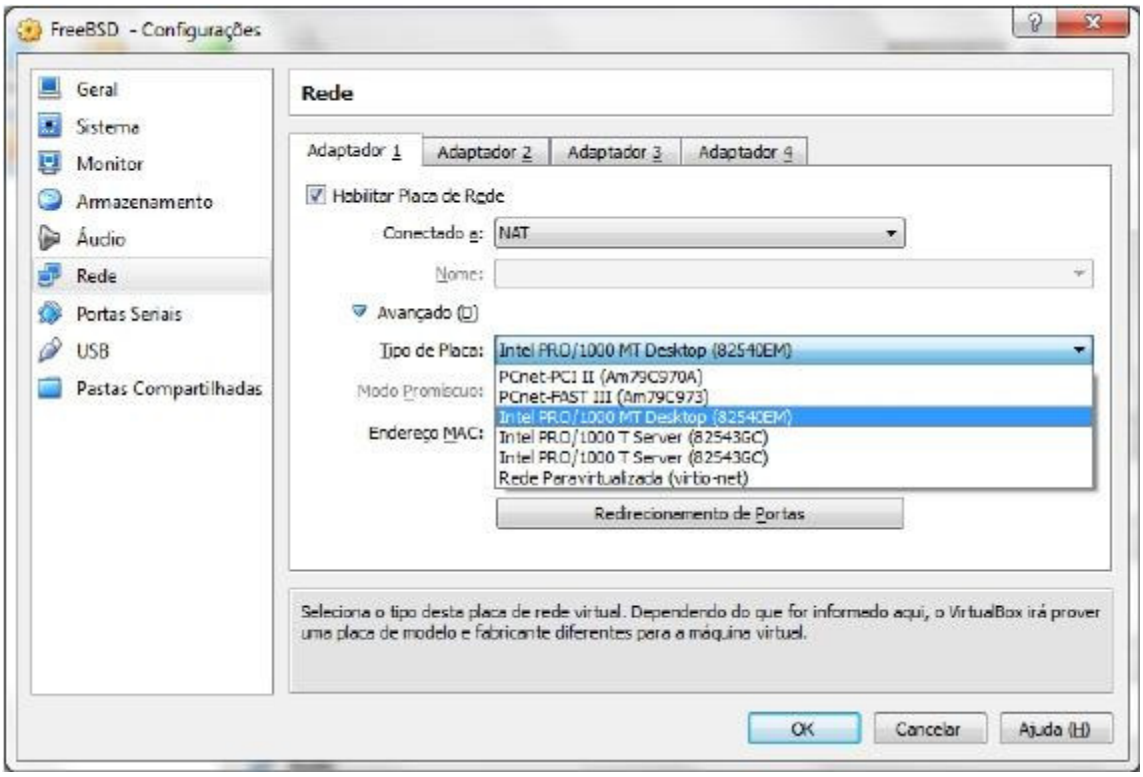
Precisamos colocar a ISO do FreeBSD no drive de CD da nossa VM. Para isso vamo no menu Armazenamento, escolhemos o drive vazio. Depois vamos nos atributos do CD para escolher o local onde de nossa ISO.



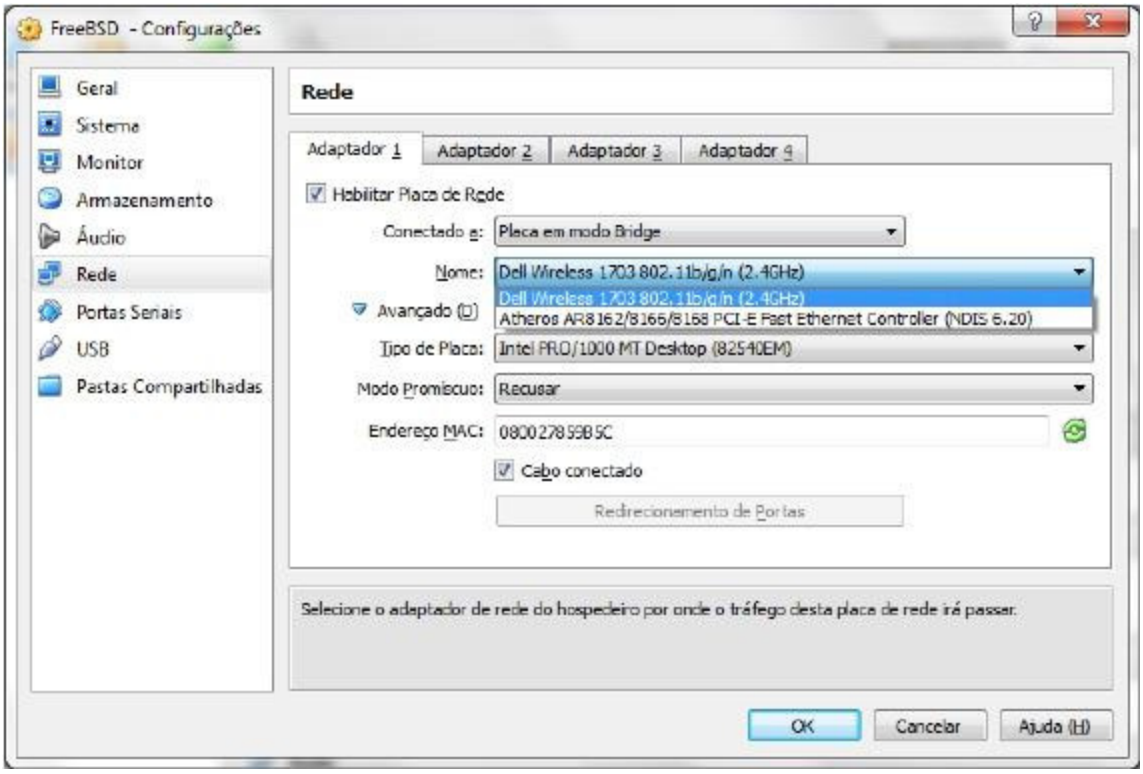
Aqui escolhemos o local onde esta a ISO.



Depois de selecionar a ISO, o drive de CD fica aparecendo o FreeBSD.



No menu Rede, vamos escolher a placa de rede que será utilizada.



Pronto a máquina esta criada e configurada. Agora vamos podemos pular para o Capítulo 3, no qual iremos falar somente de FreeBSD.

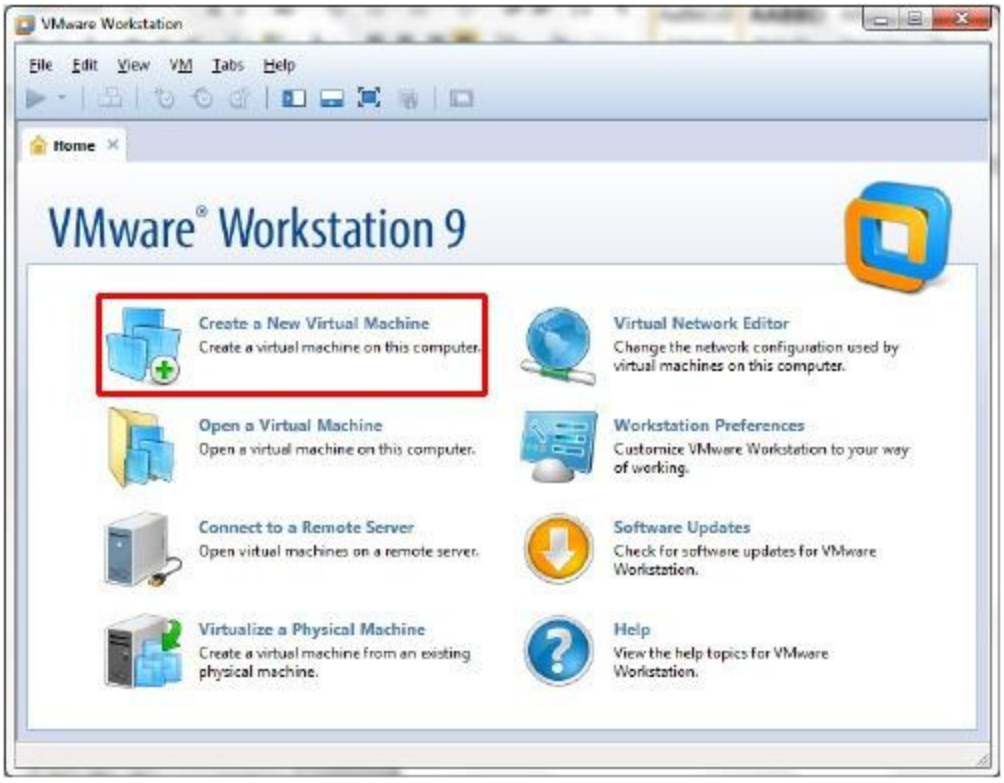
2.2. VMWARE

Atualmente a VMWARE é líder de mercado no segmento de Virtualização. Com uma infinidade de soluções para grandes DataCenters, Empresas, Instituições de Ensino, entre outras.

A VMWARE disponibiliza todas as ferramentas em seu site para download. Para algumas é preciso comprar licença de uso, para outras basta um cadastro que permite o download da aplicação.

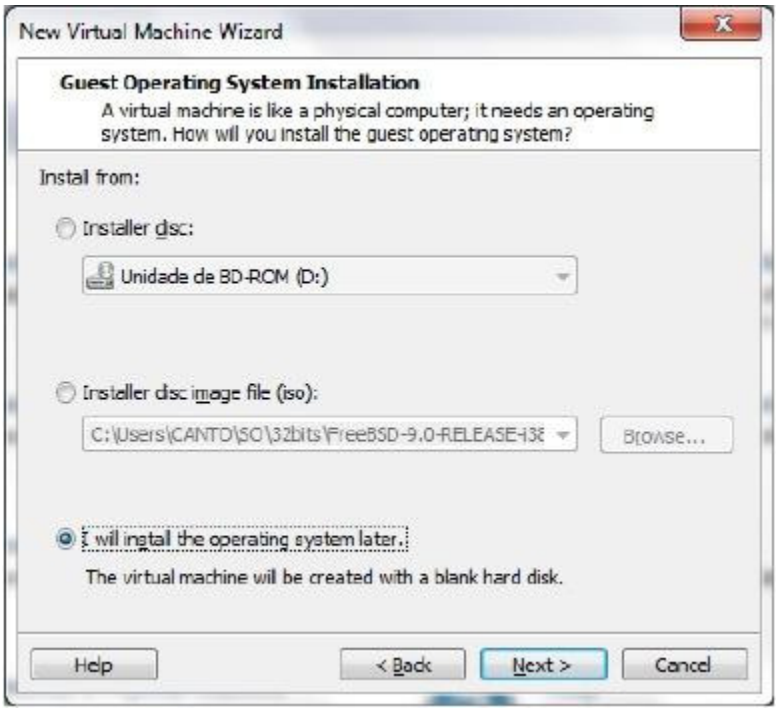
http://www.vmware.com/br/

2.2.1. Criando Máquinas Virtuais

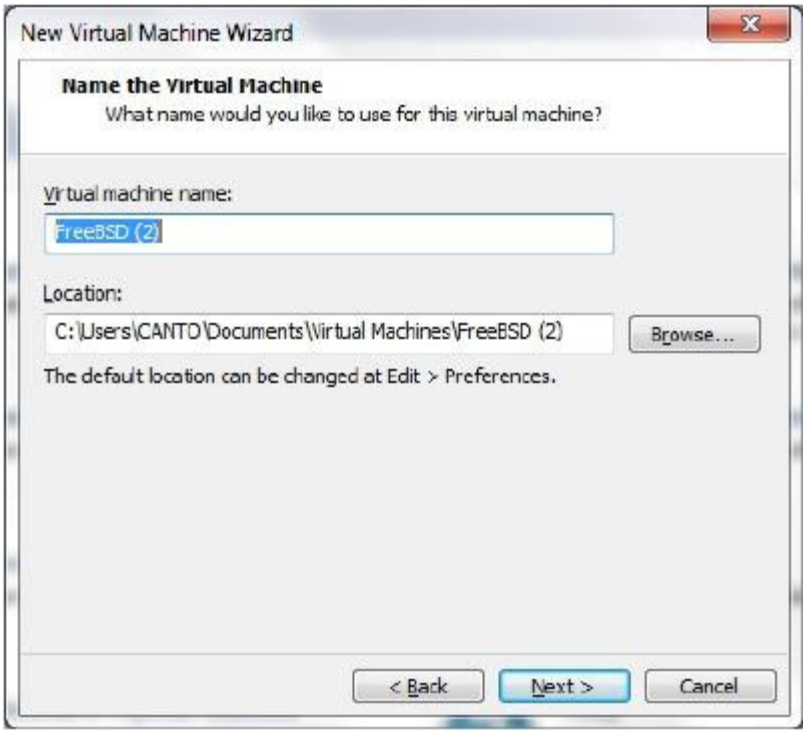


Tela inicial do VMWare Workstation. Para começar a criação de uma nova VM, basta clicar no primeiro botão – Create a New Virtual Machine.

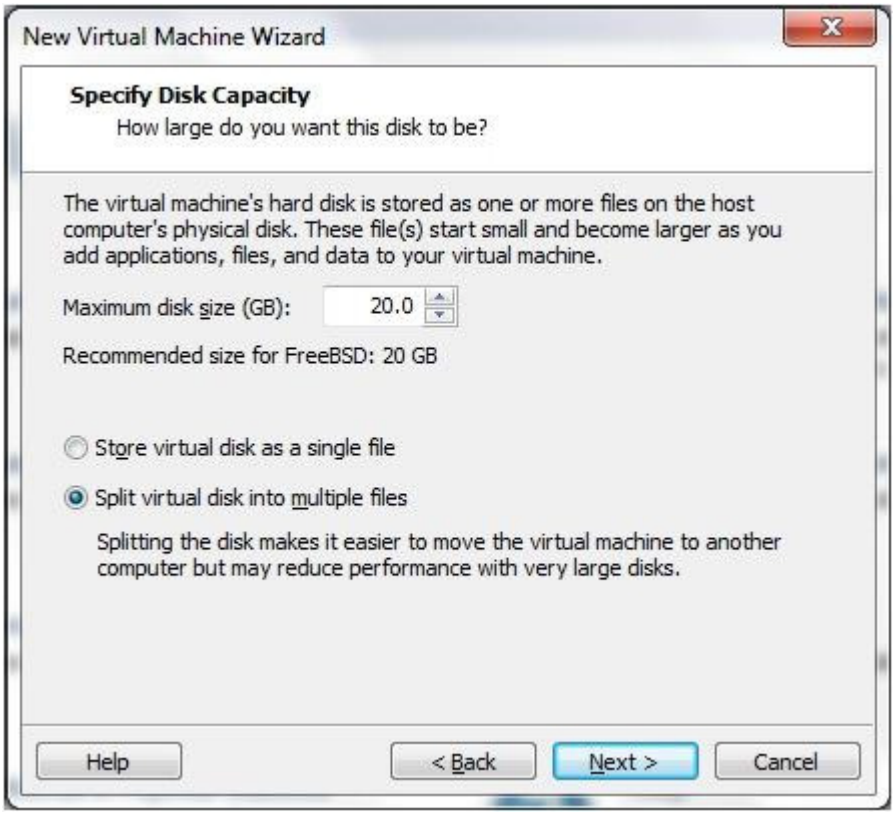
Escolha da maneira como vamos criar a VM.



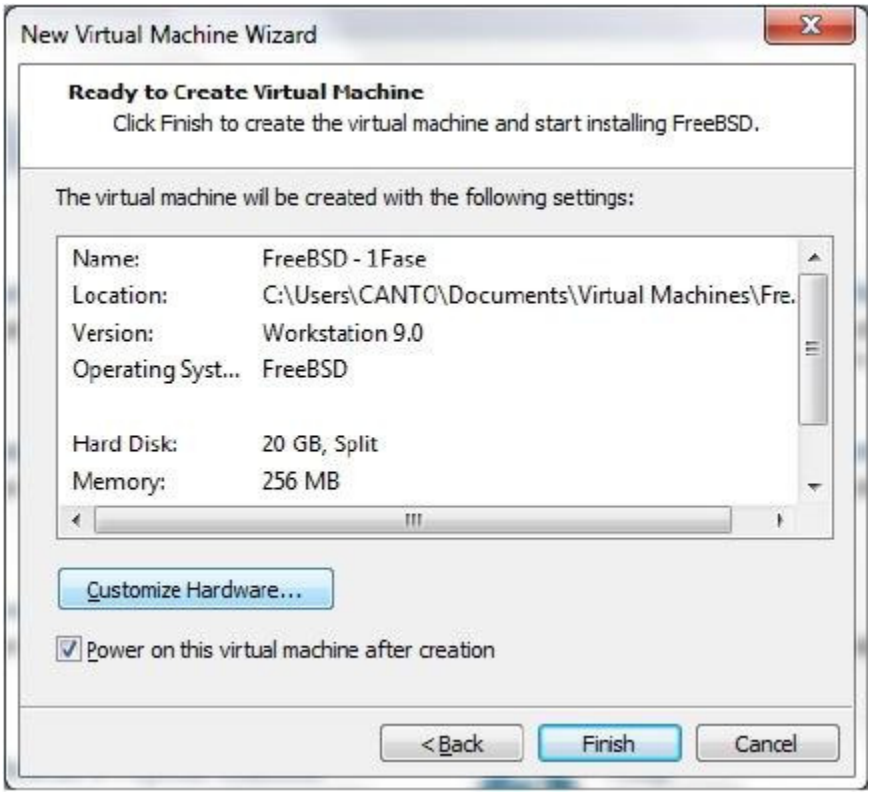
Local de onde será selecionada a instalação do S.O.



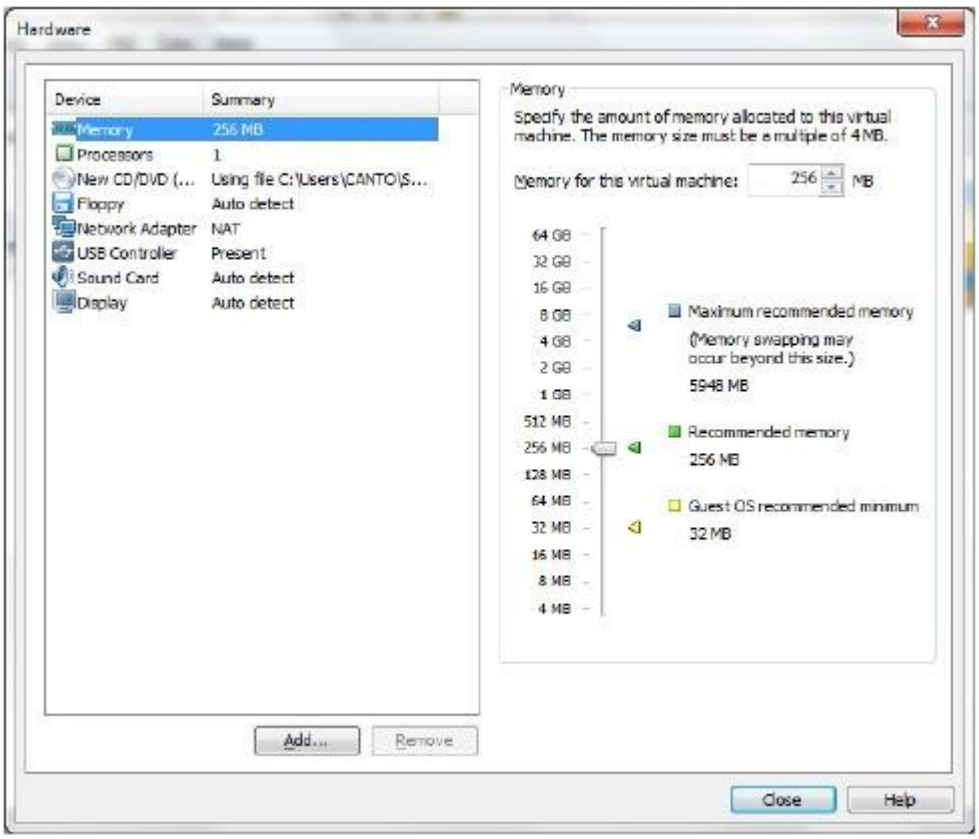
Nome da máquina e local onde ficará o arquivo principal.



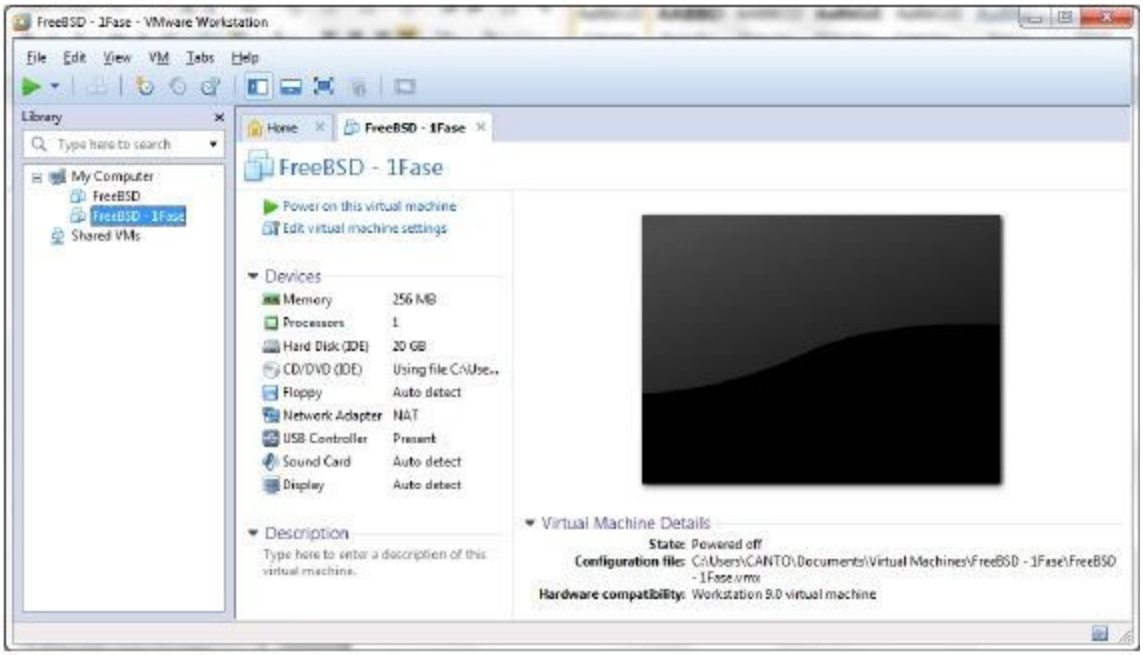
Capacidade do disco da VM.



Informações da VM. Para entrar nas configurações especificas da VM, podemos clicar no botão Customize Hardware...



Opção de customização da VM.



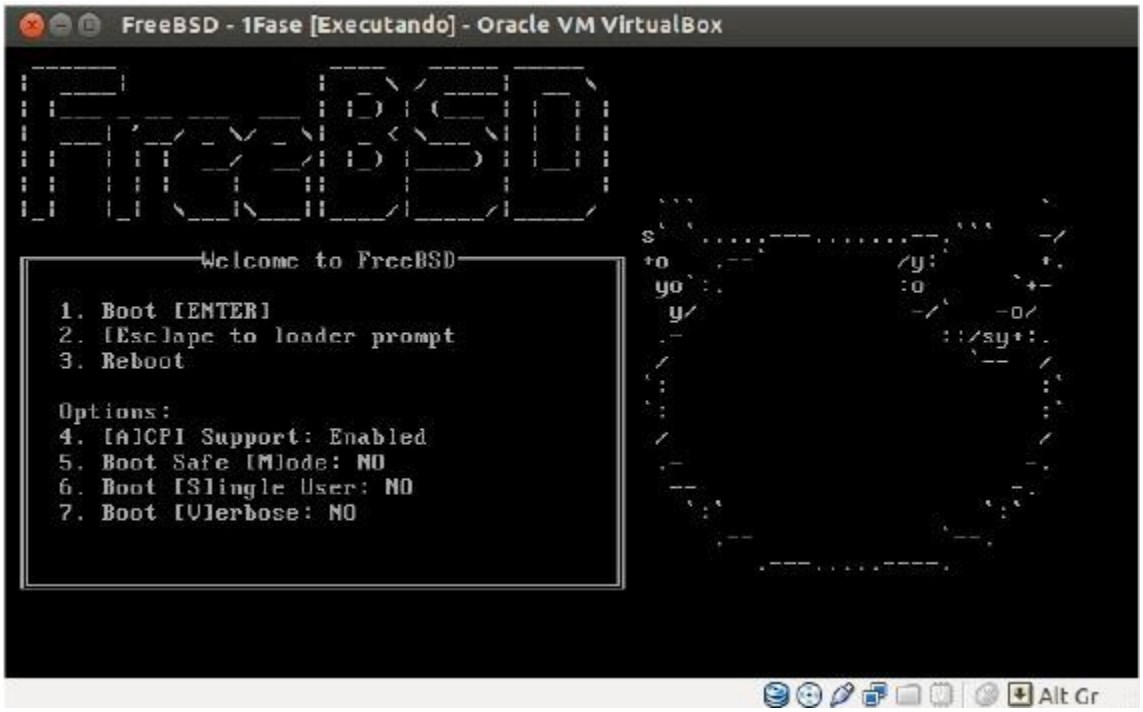
Pronto nossa VM foi criada com sucesso.

3. FREEBSD

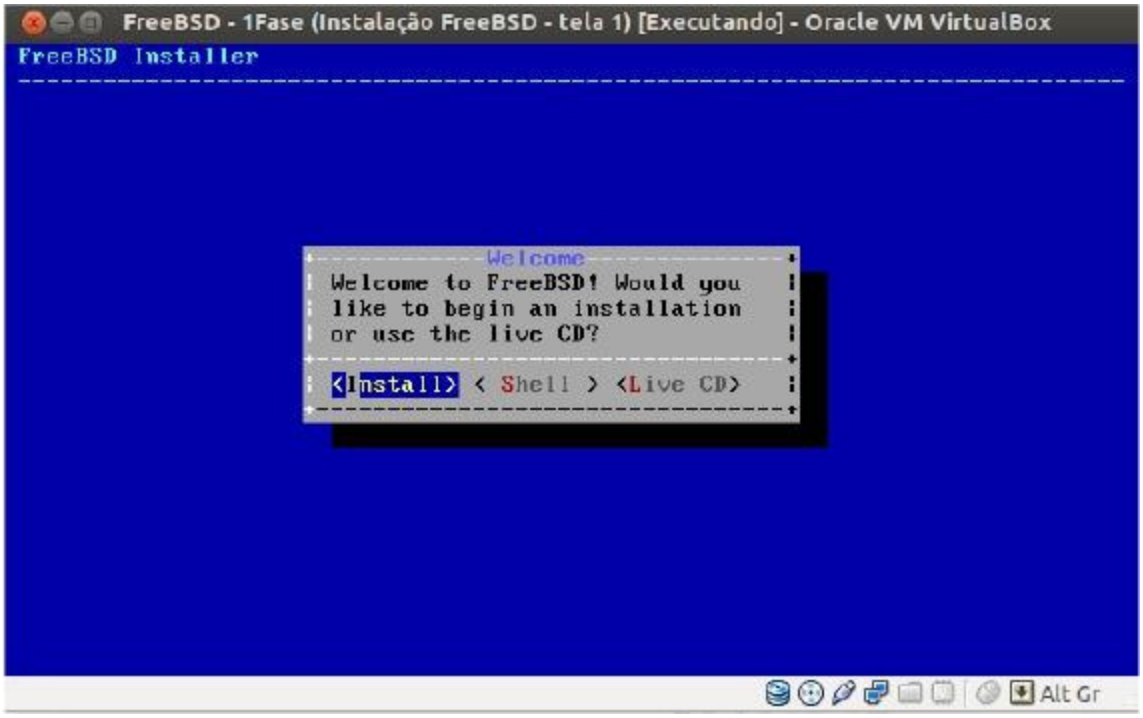
3.1. INSTALAÇÃO DO FREEBSD

Neste capítulo vamos tratar da instalação e comandos básicos para utilização do S.O. FreeBSD. Estes passos da instalação servem para qualquer Virtualizador disponível no mercado atualmente. Se em algum momento existirem diferenças entre os Virtualizadores, elas serão explicitadas no texto.

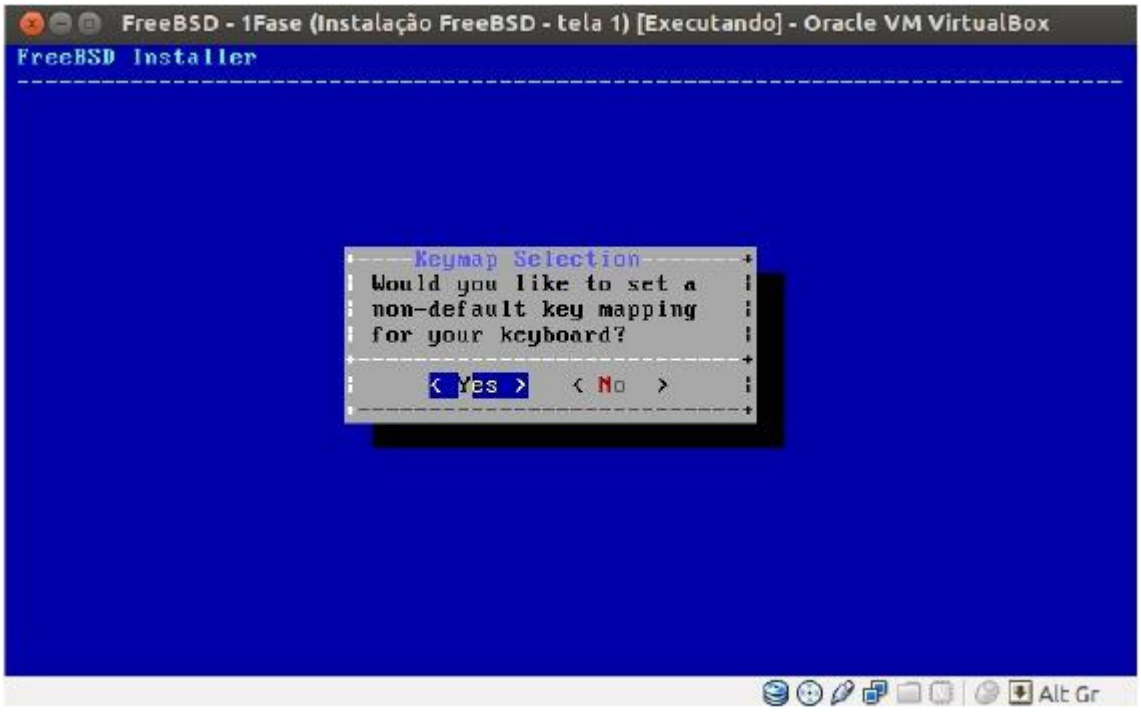
Tendo por base que o Virtualizador já esteja instalado e sua VM já esteja criada, vamos começar a instalação do FreeBSD 9.0.



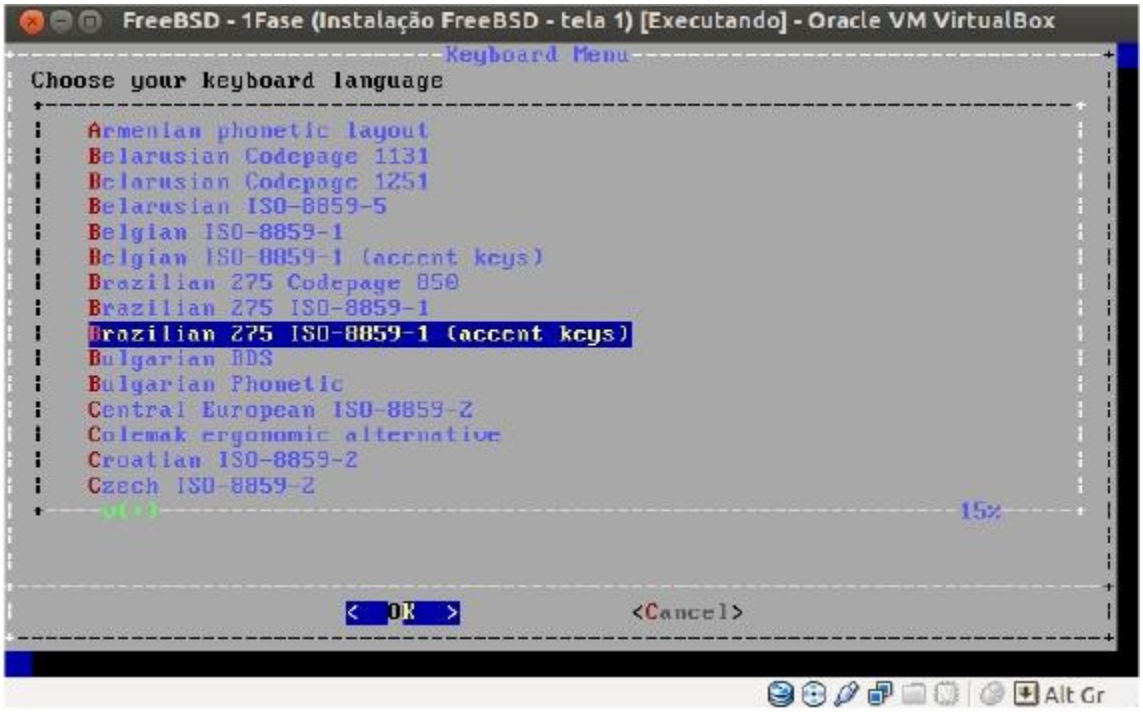
Essa é a primeira tela que vai aparecer. Nela existem algumas opções para a inicialização do S.O. que veremos no decorrer da apostila. No momento apenas vamos da um [Enter].



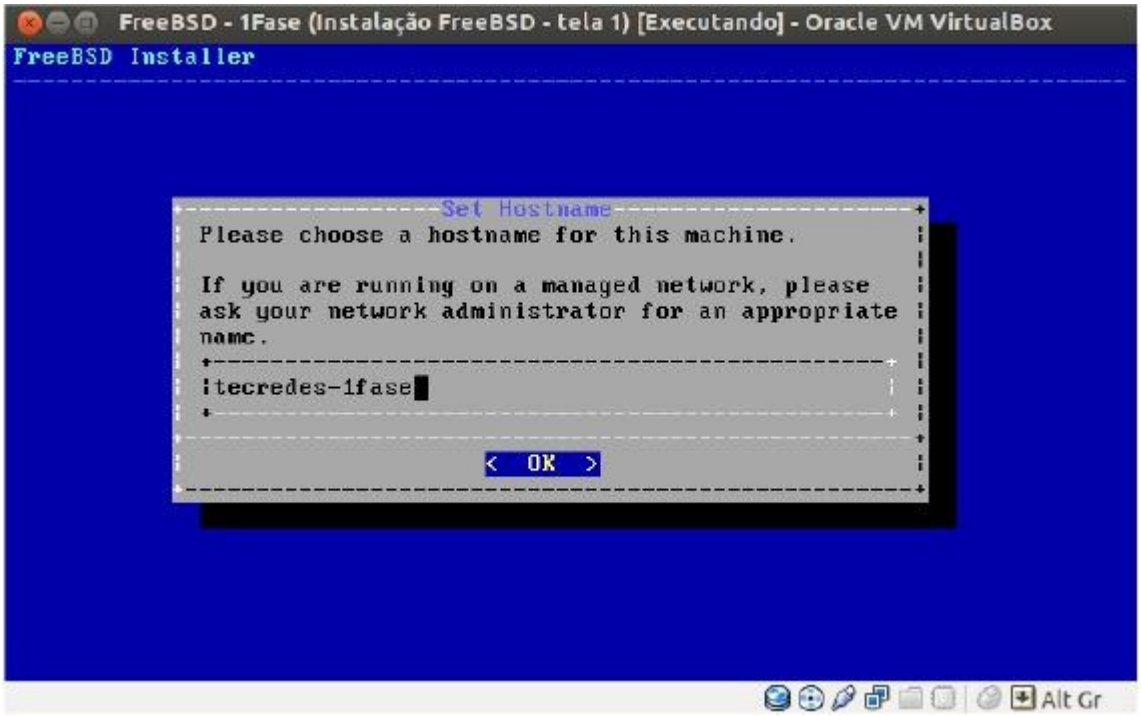
Nessa tela, já aparecem as primeiras opções para a utilização do FreeBSD. Podemos fazer a Instalação Guiada <Install>, Instalação pelo Shell <Shell> ou utilizá-lo através do próprio CD <Live CD>. Escolhemos a opção <Install>.



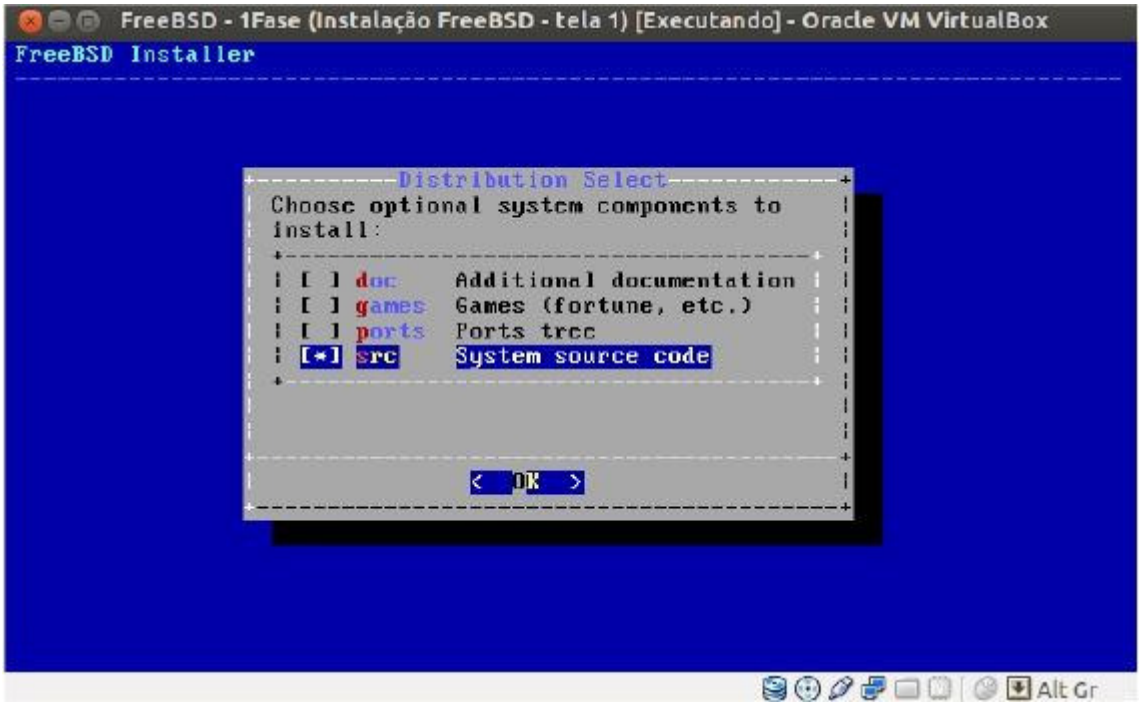
Logo em seguida, a pergunta que recebemos é para escolhermos o tipo de teclado que vamos configurar para utilizar na máquina que estamos instalando. Se for um teclado com “Ç” vamos escolher um brasileiro, conforme a próxima figura.



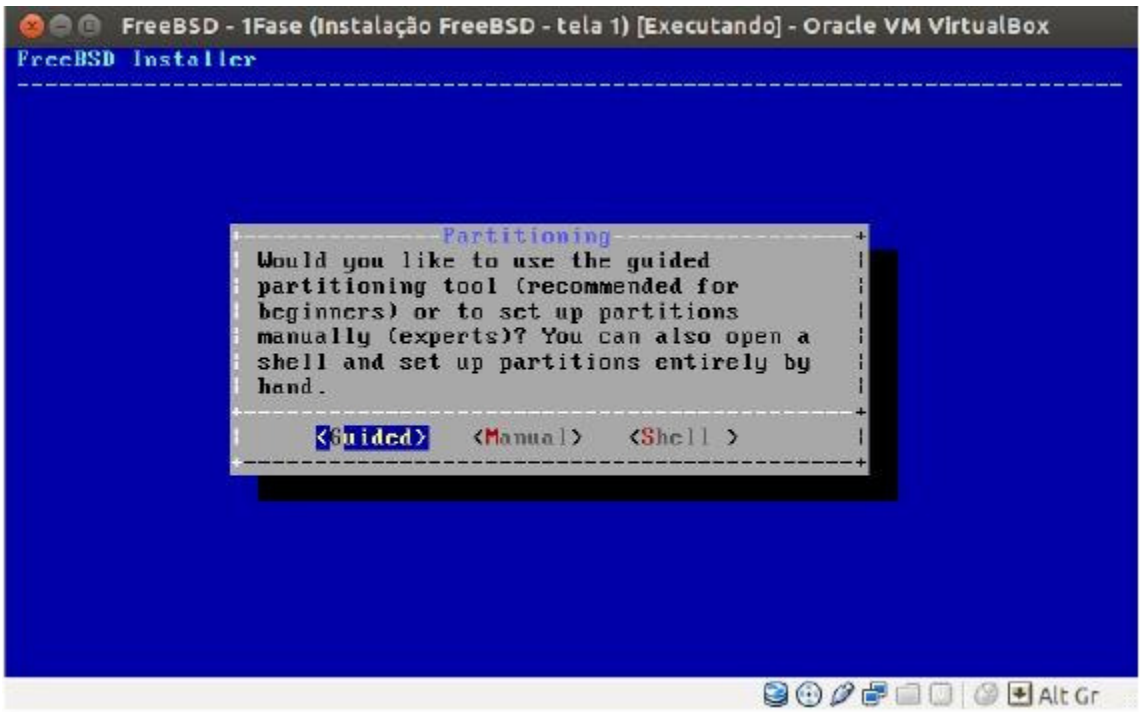
Na sequencia, vamos colocar um nome para esta máquina. Este nome pode ser o mesmo da VM, ou qualquer outro que você queira. Este será o nome da sua máquina da rede.



Devemos sempre colocar um nome que se possa identificar a máquina facilmente.



Nessa tela temos a primeira parte realmente importante para a instalação. São componentes que o S.O. utiliza. Elas podem ser instalas junto com ele, ou após a sua instalação. Os dois mais importantes são [ports] e o [src]. Porém agora só vamos escolher o [src] que são arquivos fontes do S.O. Até porque para fazer a instalação do [ports] é preciso de mais espaço em disco.



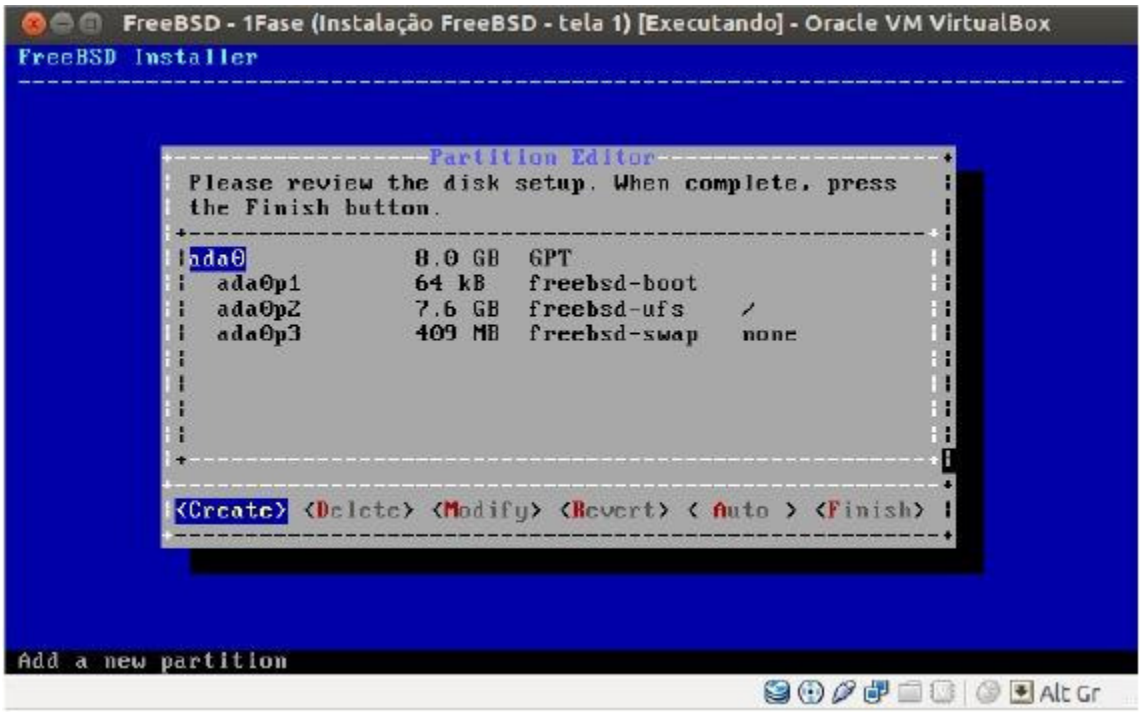
Aqui começas a parte de Particionamento de Disco. Que pode ser feita com auxilio do próprio instalador <Guided>, de forma Manual <Manual> que requer um pouco mais de conhecimento ou pelo próprio <Shell>. Vamos escolher a <Guided>.

Como não temos uma partição específica vamos utilizar o disco inteiro para o S.O., escolhendo a opção <Entire Disk>. Se fosse uma instalação em máquina física, por exemplo, podíamos escolher a opção <Partition> e definir onde iramos fazer a instalação. Mas o que é uma partição?

Partições são divisões lógicas numa mesma unidade de disco. Para que um disco seja utilizado, ele precisa primeiramente ser particionado e depois formatado, que é quando os espaços utilizáveis são alocados e endereçados. Embora fisicamente alocadas na mesma unidade, as partições comportam-se como discos rígidos independentes. Pode-se ter mais de uma partição no mesmo computador por questões de segurança, por economia de espaço em alguns Sistemas Operacionais (SO) ou para compartilhar dois SO num mesmo disco, como é o caso que deve interessar a muitos.

Alguns S.O utilizam letras para designar suas diferentes partições, como no caso da família Microsoft Windows (a: b: c: d:). Já no mundo UNIX todo o conjunto de arquivos de um diretório pode estar associado a um sistema de arquivos, como é o

caso de /usr, /tmp e assim por diante. Cada um destes sistemas de arquivos vive numa partição separada, portanto se quisermos colocar / e /usr em sistemas de arquivos separados, deveremos ter uma partição para cada um deles.

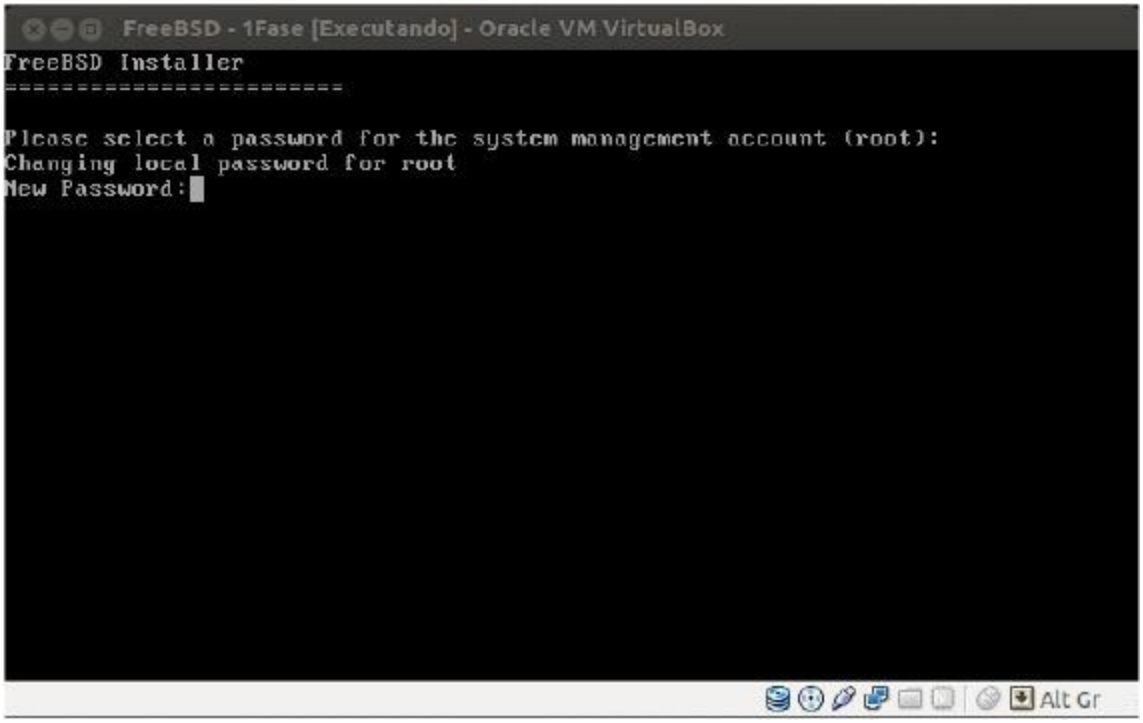


Quando optamos por fazer o Particionamento com auxílio do instalador, e já nos traz o padrão conforme a capacidade destinada para o HD da nossa VM. Podemos notar que foram criadas 3 partições, mas apenas uma é o nosso sistema de arquivos. Vamos detalhar um pouco mais o nosso Particionamento.

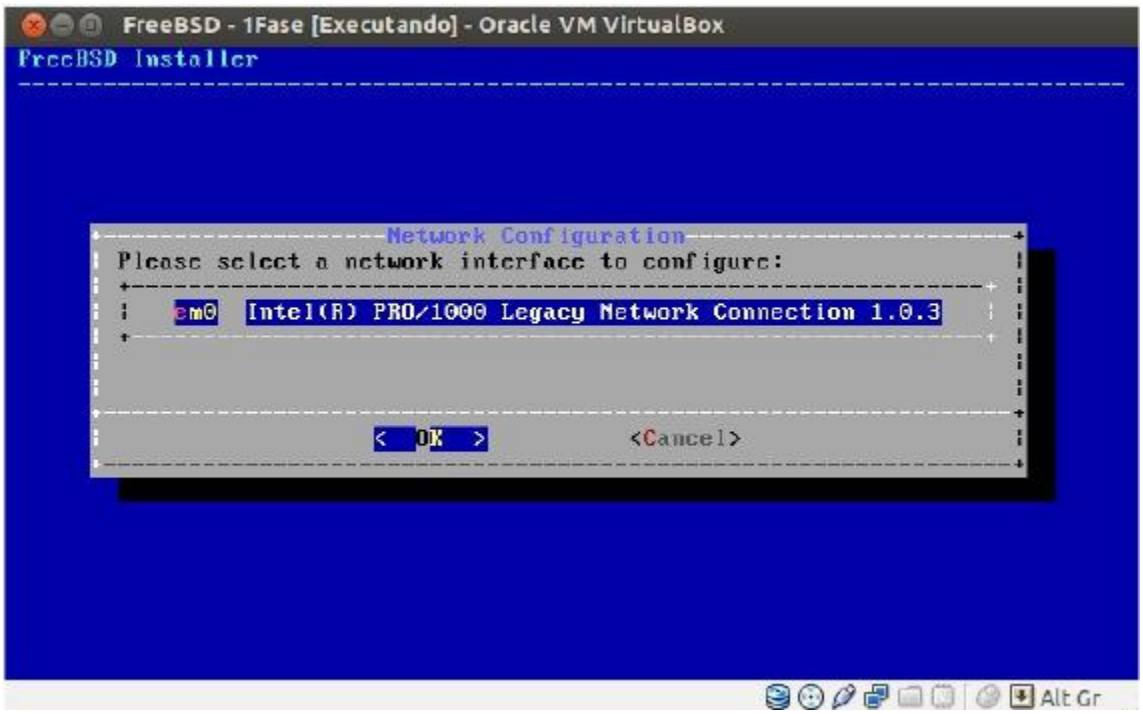
- ada0 – é o nosso HD, ou seja, o espaço que definimos na criação da VM;
- ada0p1 – é a primeira partição, espaço reservado pelo sistema para dar boot;
- ada0p2 – é partição que terá o nosso sistema de arquivos, ou seja, é o /;
- ada0p3 – por último é a swap, espaço para memória vitual.

Se quisermos fazer outra configuração de partições, podemos fazer nessa tela mesmo. Basta utilizar as opções que se encontram na parte inferior da tela. São elas: <Create>, <Delete>, <Modify>, <Revert>, <Auto> e <Finish>. O mais fácil em caso de mudanças nas partições é apaga-las e cria-las novamente.

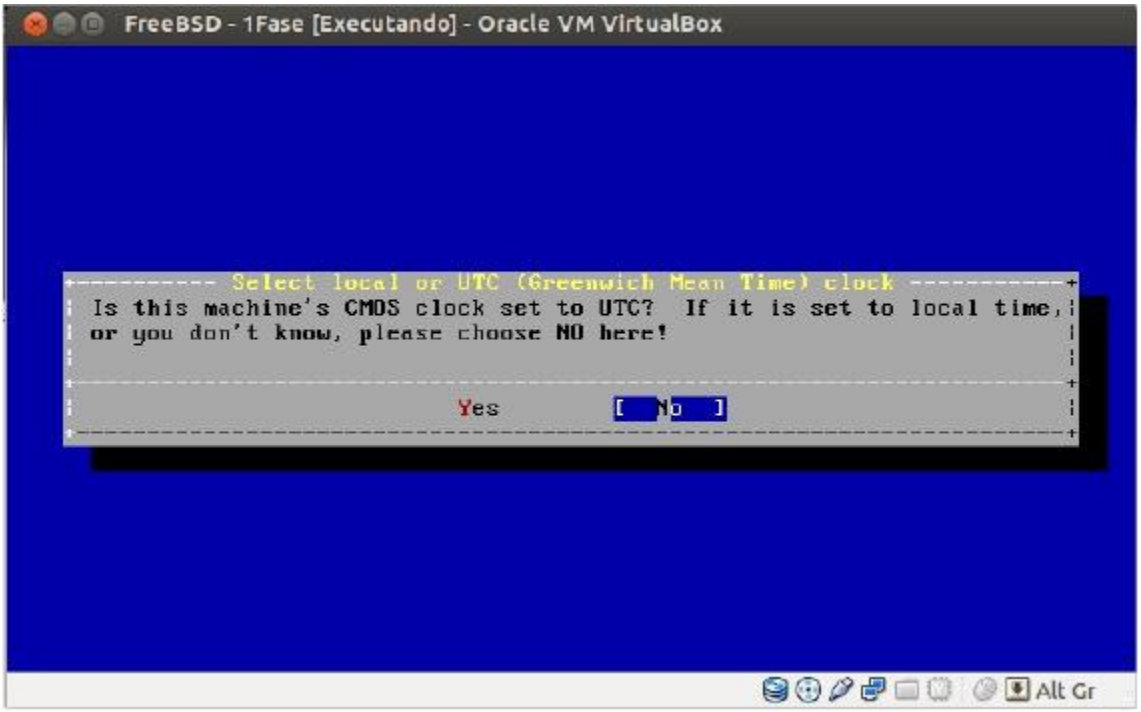
Com as partições definidas basta selecionarmos a opção <Finish> e depois <Commit>. Em seguida o instalador irá começar a extração dos arquivos para instalação.



Assim que o instalador termina a extração dos arquivos, a tela que aparece é a que está logo acima. Nela será preciso digitar uma senha. Esta senha é para o usuário root, que é o Administrador do sistema. É preciso digitar a senha duas vezes. Não se assuste, pois ao começar a digitar não aparecerão letras nem caracteres.

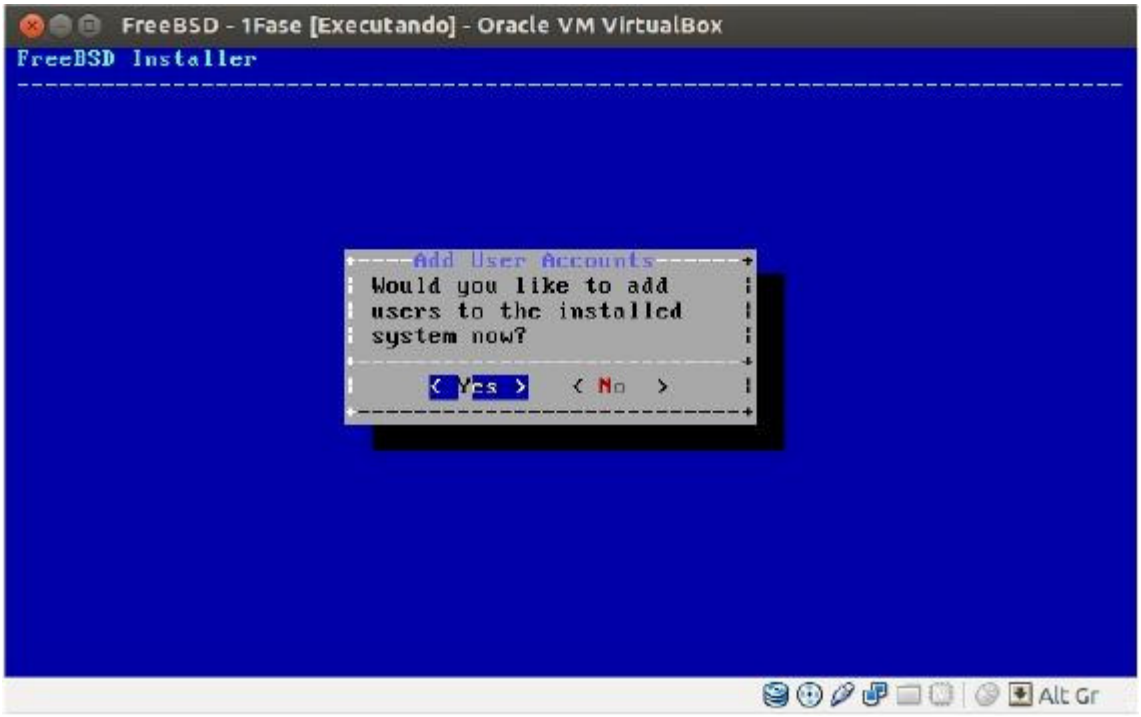


Em seguida vai aparecer a tela para selecionar a placa de rede. Basta dar <ENTER>.

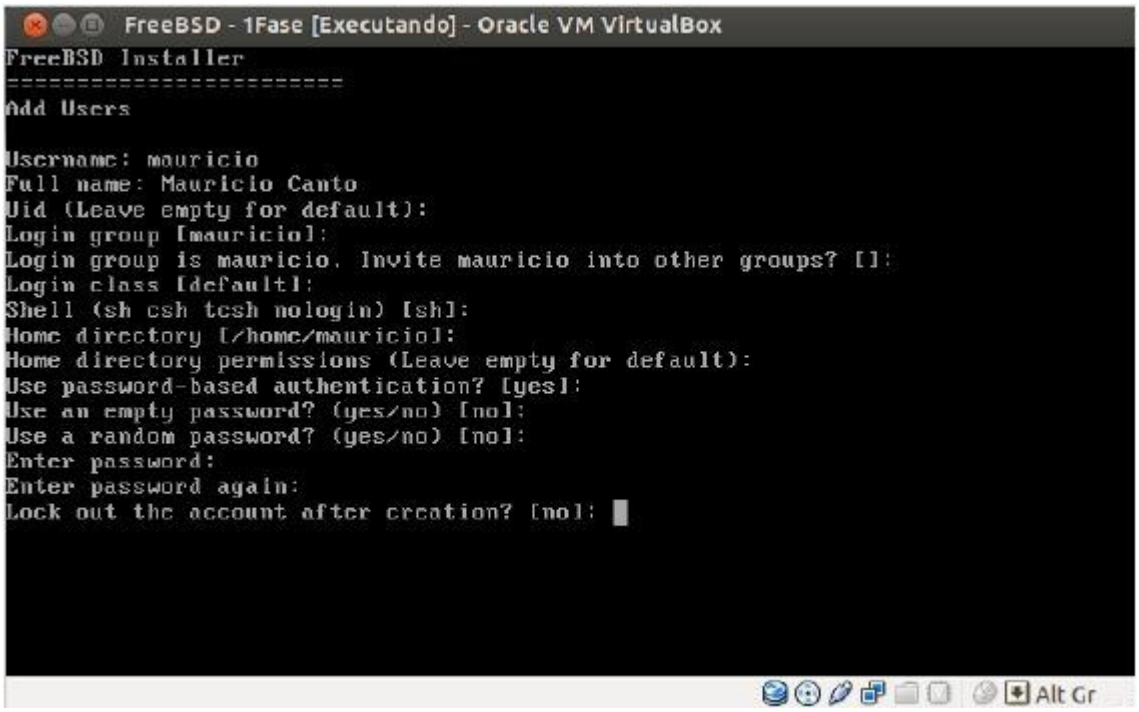


Depois de configurar a placa de rede, vamos configurar mais algumas opções do S.O. Na tela anterior vamos selecionar o fuso-horário do sistema. Para isso vamos escolher América > Brasil > Sul e Sudeste.

Após selecionar o fuso-horário vamos colocar <no> para a opção Crash dumps.



Na sequencia vamos optar por configurar mais um usuário do sistema. Ao optarmos por <yes> uma nova tela se abrirá para o cadastro de um novo usuário.



Aqui são cadastradas todas as informações para os usuários. Todos os campos que possuem informações únicas entre parentes () ou colchetes [] podem ser deixados em branco, pois as informações ali serão definidas como padrão.

• Username → Nome utilizado para se logar no sistema (login). Não é permitido espaço e é um campo casesensiteve (diferencia maiúscula de minúscula).

• Full name → Nome completo do usuário, apenas para cadastro.

• Uid → Número de identificação do usuário para o sistema.

• Login group → Grupo principal que o usuário faz parte.

• Invite mauricio into other groups? → Se deseja que o usuário faça parte de algum outro grupo.

• Login class →Restrições definias para o usuário. Elas ficam salvas no login.conf.

• Shell → Utilizamos o SH por padrão. De maneira visual todos são iguais para o usuário, mas cada um deles possuem uma estrutura e comandos próprios.

• Home directory → Diretório padrão do usuário. Por padrão todos os usuários tem seu diretório principal dentro do HOME, apenas o root tem seu diretório home e, /root.

• Home directory permissions → Definição de permissões para o diretório HOME.

• Use password-based authentication? → Utilizar senha para autenticação do usuário. SIM

• Use an empty password? → Utilizar senha vazia. NÃO

• Use a random password? → Utilizar senha aleatória. NÃO

• Enter password → a senha também é casesensitive.

• Enter password again → repita a senha para confirmar.

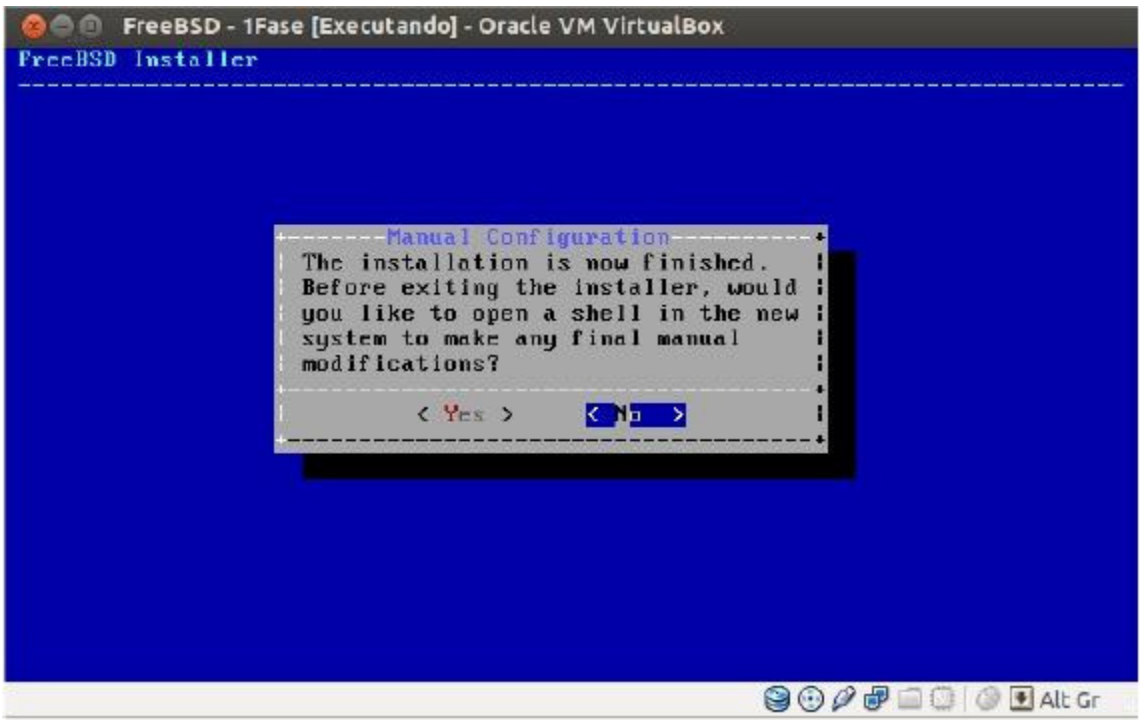
• Lock out the account after creation? → Questiona se a conta deve ficar bloqueada após a criação. NÃO.

• OK? → Após cadastrar todas as informações elas são mostradas e o sistema pergunta se elas estão corretas. É preciso digitar Y ou YES.

• Another user? → Queremos adicionar outro usuário? NÃO

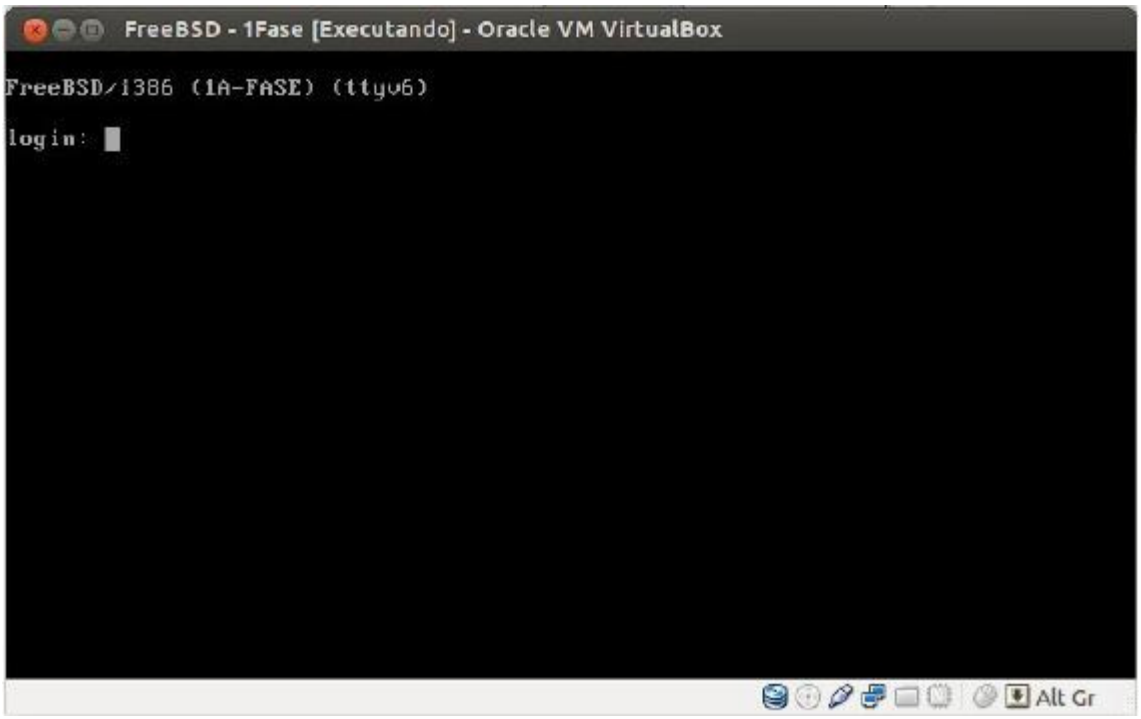
Pronto nosso sistema já esta instalado e com as contas de usuários criadas e definidas. Só precisamos concluir a instalação nas duas próximas janelas.

Nesta basta dar um <ENTER> na opção EXIT. Caso queira efetuar alguma alteração basta selecionar a opção desejada.



Por último nos vem a pergunta de desejamos efetuar alguma alteração manual antes do final da instalação. Escolhemos a opção <NO>, e depois <REBOOT>.

Agora que já instalamos a nossa VM com o FreeBSD, basta apertar no botão iniciar e começar a brincadeira.



Aqui é a tela inicial para entrarmos no FreeBSD. Basta acessar com um dos dois usuários que colocamos a senha. Um deles é o root administrador do sistema e o outro é o usuário que foi criado.

3.2. COMANDOS DE SHELL

De nada nos é útil realizar a instalação do sistema sem antes conhecer alguns comandos básicos e essenciais para o bom manuseio do FreeBSD. Mas antes de mais nada, você deve ter em mente do que é a shell? A shell é interface de comunicação entre o usuário e o sistema, disponível em modo texto e utilizada a partir de comandos. Há vários interpretadores de comando disponíveis para o FreeBSD, sendo mais conhecido o sh (que vem por padrão) e o bash (utilizado por padrão em outros sistemas, como Linux). Então fica mais fácil entender a real necessidade de lidarmos de forma prática com estes comandos.

3.2.1. Entrando e saindo do sistema

Na imagem anterior vemos primeira tela que vamos ver quando iniciarmos o sistema. Vamos acessar com os dois usuários para vermos a diferença inicial dos mesmos.

Quando entramos com root aparece a seguinte tela.

O nome da máquina, que foi hostname que cadastramos no inicia do processo de instalação, e o caractere #. É justamente esse caractere que indica que estamos utilizando o sistema com o usuário root. Vamos ver a diferença para o usuário comum. Para sairmos deste login podemos utilizar dois comandos.

- Exit / logout – encerra a sessão atual e solicita um novo login.

Entramos agora com o outro usuário que cadastramos no final da instalação. Vejamos a diferença entre a tela do root e do outro usuário.

Assim que logamos, já conseguimos ver a diferença entre os usuários. Aquele caractere # que aparece para o usuário root, foi substituído pelo $. Dessa forma já identificamos que tipo de usuário estamos logados.

Para sair e reiniciar o sistema existem alguns comandos. Porém, por questões de segurança esses comandos só podem ser executados pelo usuário root.

- Reboot – serve para reiniciar o sistema.
- Halt – serve para desligar o sistema.
- Shutdown -h now – serve para desligar o sistema e informar outros usuários que estejam logados que o sistema esta sendo desligado.
- init 0 – serve para encerrar e matar todos os processos antes de desligar o sistema.

Como podemos observar na imagem anterior, percebemos que o usuário comum não possui permissão para executar nenhum dos comandos acima. Para utilizá-los precisamos nos logar novamente como root. Para isso vamos utilizar o comando login. Com este comando o sistema solicita que digite a o usuário e senha, mas ele não encerra a sessão que estávamos anteriormente.

- login – abre uma nova sessão sem encerrar a anterior.

3.2.2. Comandos, manuseio de arquivos e diretórios

Todos os comandos possuem parâmetros para melhor utilização dos mesmos. E por isso eles possuem um manual que explica a forma de utilização do comando e seus parâmetros.

- Man – abre o manual de utilização do comando.

- pwd - print working directory (mostrar o diretório atual). Mostra a pasta na qual o usuário está naquele momento. Bastante útil para nos situarmos em que lugar estamos no sistema.

```
1A-FASE# pwd
/root
1A-FASE#
```

- cd - Command directory (Comando de diretório). O cd permite ao usuário navegar entre as pastas, de forma bastante simples.

Entrou na pasta teste .....

```
1A-FASE# pwd
/root
1A-FASE# cd teste/
1A-FASE# pwd
/root/teste
1A-FASE#
```

Volta à pasta anterior .....

```
1A-FASE# pwd
/root/teste
1A-FASE# cd ..
1A-FASE# pwd
/root
1A-FASE#
```

Vai à pasta inicial raiz do sistema .....

```
1A-FASE# pwd
/root
1A-FASE# cd /
1A-FASE# pwd
/
1A-FASE#
```

Vai à pasta inicial do usuário , normalmente /home/usuário ...

```
1A-FASE# pwd
/
1A-FASE# cd
1A-FASE# pwd
/root
1A-FASE#
```

Navega direto à pasta desejada .....

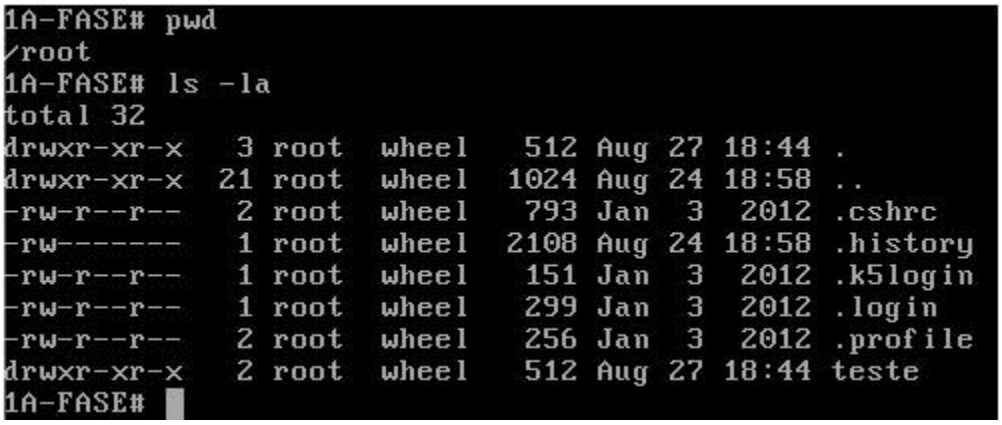
```
1A-FASE# cd /etc/ssh/
1A-FASE# pwd
/etc/ssh
```

No caso da não existência da pasta, o sistema avisa ao usuário do problema, normalmente com uma mensagem dizendo que o diretório não existe.

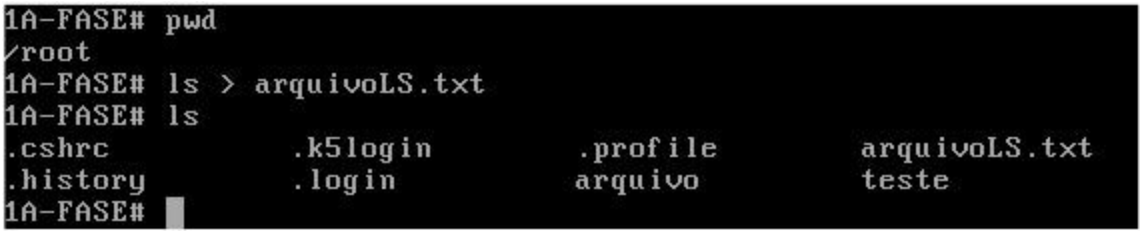
```
1A-FASE# cd /teste/teste2
/teste/teste2: No such file or directory.
1A-FASE#
```

- ls - Listen (Listar). O ls serve basicamente para listar arquivos em um diretório, possuindo como parâmetros algumas opções, como o -la.

Exemplo de uso: Listar as pastas e arquivos existentes no diretório /root



Com o parâmetro -la, um grupo de informações interessantes, tais como a permissão do arquivo ou pasta, o dono e o grupo do arquivo, e a data de criação/modificação. Pode-se ainda modificar a saída do comando ls para um arquivo, através do comando ls > "nome do arquivo".



Pode-se também usar no ls caracteres que substituem outros como, por exemplo, o (*). Caso o usuário queira listar todos os arquivos com a terminação ".txt" fará o seguinte:

- mkdir - Make Dir (Criar diretório). Cria uma pasta ou diretório, como é mais conhecido. Basta usar simplesmente o comando mkdir nome da pasta, que ela é criada.

- rmdir - Remove Dir (Remover diretórios) . Comando utilizado apenas para remover diretórios.

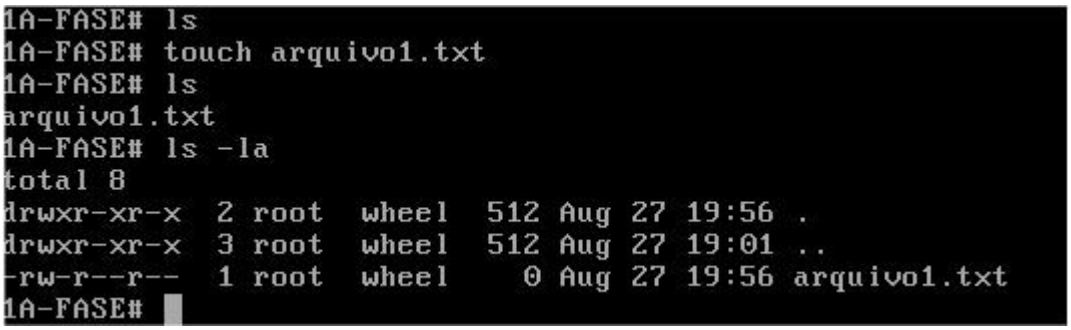
- rm - Remove (Remover arquivos e diretórios) Deleta um arquivo ou pasta. Pode-se utilizar com o parâmetro -rf, que obriga o arquivo ou pasta a ser deletado. O parâmetro r significa recurse (recursivo) e f é force (forçar). Então utilizando o parâmetro rf o arquivo ou pasta será apagado de forma obrigatória. Muito cuidado ao utilizar este parâmetro, pois ele irá apagar todos os arquivos e diretórios que estiverem dentro do mesmo.

Exemplificando:

Utilizando somente o rm em um diretório, ele dará um erro informando que isto é um diretório.

Só conseguimos apagar um diretório utilizando os parâmetros -rf.

- touch - tocar, criar. Cria um arquivo em branco, com tamanho zero.



- ee - easy editor (editor fácil). Para iniciar o ee, basta digitar na linha de comando ee <nomedoarquivo> onde nomedoarquivo é o arquivo que se quer criar ou editar.

Depois de entrar no ee é só começar a escrever o conteúdo que precisamos. Na parte superior da tela estão os comandos que podem ser utilizados dentro do Editor. O sinal ^ significa a utilzação da tecla Ctrl. Para salvar e sair do arquivo utilizamos a tecla Esc.

- cat - (Exibir). Exibe um arquivo na tela ou outra saída de comando. O cat é bastante útil ao usuário pelo fato de não precisar especificamente abrir um arquivo para listar o seu conteúdo dentro. Usa-se o cat em sistemas FreeBSD principalmente para listar arquivos de configuração do sistema.

Exemplificando:

```
1A-FASE# ls
arquivo1.txt
1A-FASE# cat arquivo1.txt
aqui podemos escrever o
conteudo do arquivo

---------------
##### FIM #####
---------------
1A-FASE#
```

- less – (Exibir com paginação). O comando less do sistema operacional Unix serve para mostrar arquivos texto. Possui a capacidade de exibir o arquivo de maneira paginada, com opção de rolagem para trás e para frente. Como não necessita ler todo o arquivo de entrada antes de exibi-lo, consegue maior desempenho comparado a outros visualizadores e editores quando manipulam arquivos grandes.

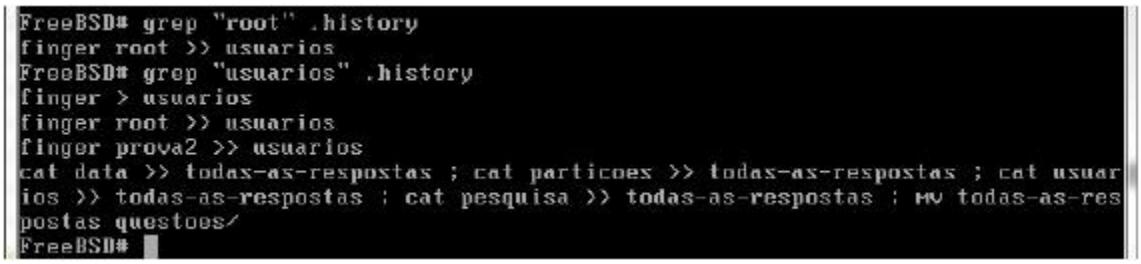
```
sshd:*:22:22:Secure Shell Daemon:/var/empty:/usr/sbin/nologin
smmsp:*:25:25:Sendmail Submission User:/var/spool/clientmqueue:/usr/sbin/nologin
mailnull:*:26:26:Sendmail Default User:/var/spool/mqueue:/usr/sbin/nologin
bind:*:53:53:Bind Sandbox:/:/usr/sbin/nologin
proxy:*:62:62:Packet Filter pseudo-user:/nonexistent:/usr/sbin/nologin
_pflogd:*:64:64:pflogd privsep user:/var/empty:/usr/sbin/nologin
_dhcp:*:65:65:dhcp programs:/var/empty:/usr/sbin/nologin
uucp:*:66:66:UUCP pseudo-user:/var/spool/uucppublic:/usr/local/libexec/uucp/uucico
pop:*:68:6:Post Office Owner:/nonexistent:/usr/sbin/nologin
www:*:80:80:World Wide Web Owner:/nonexistent:/usr/sbin/nologin
hast:*:845:845:HAST unprivileged user:/var/empty:/usr/sbin/nologin
/etc/passwd
```

Com ele podemos rolar o conteúdo do arquivo com aas setas de direcionamento e também com as teclas Page Down e Page Up.

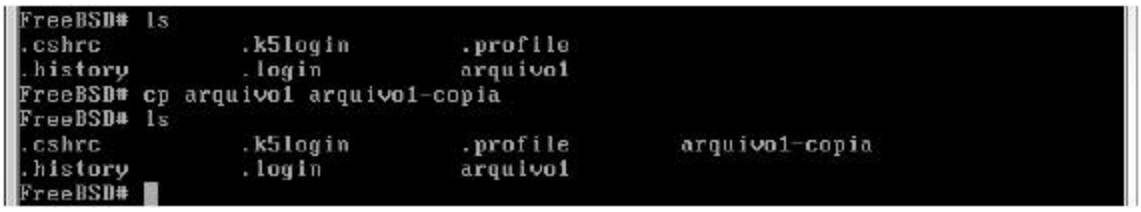
- more – faz uma pausa entre a exibição de uma tela e outra.

```
#1354922717
mkdir questoes ; touch todas-as-respostas
#1354922743
finger > usuarios
#1354922753
finger root >> usuarios
#1354922763
finger prova2 >> usuarios
#1354922802
find / -name "*.log" > pesquisa
#1354922819
date >> data
#1354922908
cat data >> todas-as-respostas ; cat particoes >> todas-as-respostas ; cat usuarios >> todas-as-respostas ; cat pesquisa >> todas-as-respostas ; mv todas-as-respostas questoes/
#1354922910
ll
--More--(45%)
```

- grep – procura uma palavra dentro de um arquivo. Funciona também aninhado com outros comandos.



- cp – copia arquivos e diretórios. No exemplo abaixo, temos o arquivo1 e vamos criar uma copia dele, chamada de arquivo1-copia. A utilização deste comando deve ser a seguinte: cp <nome do arquivo original> <nome da cópia>.



O arquivo copiado terá exatamente o mesmo conteúdo que o arquivo original. É a mesma coisa que utilizarmos o CTRL+C e CTRL+V do Windows.

- mv – mover ou renomear arquivos e entre diretórios.

Podemos utilizar o comando mv tanto para mudar um arquivo de diretório como para renomear um arquivo. No exemplo abaixo renomeamos o arquivo1 para arquivo1-renomeado.

Já no próximo exemplo, movemos o arquivo1-renomeado, que estava dentro do diretório /root para o diretório /root/novo.

```
FreeBSD# mkdir novo
FreeBSD# ls
.cshrc                  .login                  arquivo1-renomeado
.history                .profile                novo
.k5login                arquivo1-copia
FreeBSD# mv arquivo1-renomeado novo/
FreeBSD# ls novo/
arquivo1-renomeado
FreeBSD#
```

3.2.3. Informações do sistema

- df - Disk Free (Disco livre).

O df mostra o espaço disponível em disco. Pode ser utilizado com o parâmetro -H que mostra o tamanho em megabytes. Além disso, ele nos traz a partição montada, o espaço usado, disponível, capacidade utilizada e o ponto de montagem.

```
1A-FASE# df
Filesystem  1K-blocks    Used   Avail Capacity  Mounted on
/dev/ada0p2   7836636 1234752 5974956    17%    /
devfs               1       1       0   100%    /dev
1A-FASE#
```

```
1A-FASE# df -h
Filesystem     Size    Used   Avail Capacity  Mounted on
/dev/ada0p2    7.5G    1.2G    5.7G    17%    /
devfs          1.0k    1.0k      0B   100%    /dev
1A-FASE#
```

- du – Informa o uso do sistema em bytes

O comando abaixo mostra o tamanho dos diretórios dentro do diretório atual.

```
FreeBSD# du -h
4.0k    ./novo
 28k    .
```

- date – Data e hora atual do sistema

```
FreeBSD# date
Wed Jan 30 19:13:19 BRST 2013
```

- cal – Traz o calendário do mês corrente

```
FreeBSD# cal
    January 2013
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
```

- uname – Mostra o nome do sistema e com o parâmetro –a, traz informações completas da versão instalada.

```
FreeBSD# uname
FreeBSD
FreeBSD# uname -a
FreeBSD FreeBSD 9.0-RELEASE FreeBSD 9.0-RELEASE #0: Tue Jan  3 07:15:25 UTC 2012
    root@obrian.cse.buffalo.edu:/usr/obj/usr/src/sys/GENERIC  i386
FreeBSD#
```

- ifconfig – mostra as interfaces de rede na máquina.

```
FreeBSD# ifconfig
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=9b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM>
        ether 00:0c:29:fa:ed:f3
        inet 192.168.116.128 netmask 0xffffff00 broadcast 192.168.116.255
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
plip0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> metric 0 mtu 1500
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x5
        inet 127.0.0.1 netmask 0xff000000
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
FreeBSD#
```

- ln – link simbólico. Funciona como a criação de um atalho no Windows.

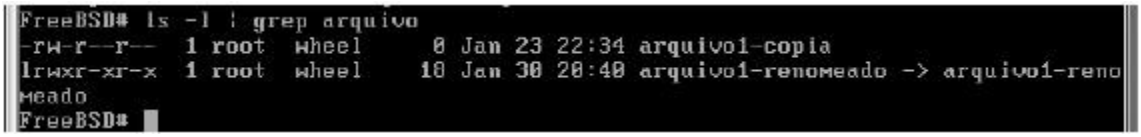
```
FreeBSD# ll
total 0
lrwxr-xr-x  1 root  wheel  14 Jan 30 20:40 arquivo1-copia -> arquivo1-copia
-rw-r--r--  1 root  wheel   0 Jan 23 22:34 arquivo1-renomeado
FreeBSD# ln -s arquivo1-renomeado ../
FreeBSD# cd ..
FreeBSD# ll
total 24
-rw-r--r--  2 root  wheel   793 Jan  3  2012 .cshrc
-rw-------  1 root  wheel  1156 Jan 30 19:02 .history
-rw-r--r--  1 root  wheel   151 Jan  3  2012 .k5login
-rw-r--r--  1 root  wheel   299 Jan  3  2012 .login
-rw-r--r--  2 root  wheel   256 Jan  3  2012 .profile
-rw-r--r--  1 root  wheel     0 Jan 23 22:34 arquivo1-copia
lrwxr-xr-x  1 root  wheel    18 Jan 30 20:40 arquivo1-renomeado -> arquivo1-reno
meado
drwxr-xr-x  2 root  wheel   512 Jan 30 20:40 novo
FreeBSD#
```

3.2.4. Redirecionamentos

O shell nos permite redirecionar as saídas (outputs) de comandos para outros comandos ou para arquivos. Isso nos proporciona inúmeras possibilidades de aninhamento de comandos.

- Pipe

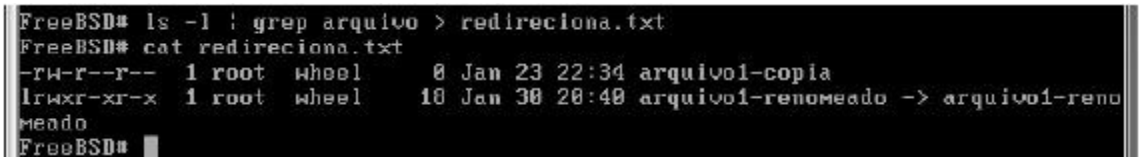
O pipe "|" serve para passar o output de um comando para outro, ou seja antes de passar para a tela o resultado você pode literalmente redirecionar para onde vai seu comando.



No exemplo acima, temos a utilização de dois comandos. O ls, que o seu resultado seria listar todos os arquivos e diretórios de onde estamos e o grep, que faz um filtro pela palavra que definimos. Como juntamos os dois o que o resultado do ls junto com o grep foi a listagem de arquivos e diretórios que continha a palavra definida.

- >

O símbolo de maior ">" redireciona a saída de um comando para dentro de um arquivo. Veremos melhor o seu funcionamento no exemplo abaixo.



Utilizando o mesmo comando do exemplo anterior, apenas adicionamos a símbolo de > para que o resultado do comando seja salvo no arquivo redireciona.txt. Logo em seguida é mostrado o conteúdo do arquivo. É exatamente o mesmo que foi feito no item anterior.

Importante: se este símbolo for utilizado com algum arquivo existente, o seu conteúdo será reescrito.

- >>

O funcionamento dos dois sinais de maior juntos “>>”, é muito parecido com o outro. A única diferença é que esse não sobrescreve o conteúdo existente no arquivo. Ele sempre vai adicionando o conteúdo no final.

Vamos utilizar o mesmo arquivo anterior para fazer um teste.

```
FreeBSD# echo "Adicionando texto no arquivo redireciona.txt" >> redireciona.txt

FreeBSD# cat redireciona.txt
-rw-r--r--  1 root  wheel     0 Jan 23 22:34 arquivo1-copia
lrwxr-xr-x  1 root  wheel    18 Jan 30 20:40 arquivo1-renomeado -> arquivo1-reno
meado
Adicionando texto no arquivo redireciona.txt
FreeBSD#
```

3.2.5. Localizando arquivos.

Os sistemas Unix possuem alguns comandos diferentes que podem ser utilizados na procura de arquivos e programas. Cada um deles possui suas vantagens e desvantagens. Vamos conhecer um pouco de cada um.

- whereis - busca por arquivos executáveis, man pages, arquivos de configuração e fontes.

```
FreeBSD# whereis ls
ls: /bin/ls /usr/share/man/man1/ls.1.gz
FreeBSD#
```

O resultado da procura nos trouxe o arquivo executável do comando e também o manual de utilização do mesmo.

- which - busca por executáveis.

```
FreeBSD# which ls
/bin/ls
FreeBSD#
```

Estes dois comandos são os mais simples para serem utilizados, mas talvez eles não tragam o resultado esperado pelo usuário. Por isso existem comandos com uma complexidade de utilização um pouco maior. São eles o locate e o find.

- locate – busca arquivos dentro de uma base de dados própria.

O comando locate possui uma base de dados própria onde ele faz a sua pesquisa. Porém esta base de dados não vem atualizada. Portanto se fizermos uma pesquisa com este comando, receberemos como resultado que a base do locate esta vazia.

Essa é a principal desvantagem do comando. Pois a base de dados do mesmo precisar ser atualizada com certa frequência. Veremos na próxima imagem como atualizar a base de dados e a utilização do comando.

- find – faz a pesquisa no próprio sistema através de parâmetro.

O find é o comando mais chato para ser utilizado. Porém ele pode ser utilizado em qualquer situação de procura. Com ele podemos procurar arquivos, diretórios, textos dentro de arquivos, arquivos de uma determinada data, por tipo e também por tamanho.

O segredo do comando find esta na forma como ele é utilizado.

COMANDO: find </diretório/> -parametro <pesquisa>.

Esta é a forma de utilização do comando. Como se trata de um comando bem completo ele possui dezenas de parâmetros. Vamos ao mais simples.

O parâmetro –name é utilizado para pesquisar arquivos. No exemplo acima foi feita a pesquisa a partir do diretório /root/.

3.3. ESTRUTURA DE DIRETÓRIOS DO FREEBSD

O FreeBSD é sem duvida, um sistema organizado quando lidamos com a sua estrutura de diretórios. Desde o começo do desenvolvimento do sistema, há uma clara preocupação em manter a organização, facilitando assim a compatibilidade dos programas e a facilidade em encontrar os arquivos de configuração do sistema e demais aplicativos instalados. O usuário tem que entender exatamente o porque do particionador realizar de forma automática o trabalho de dividir o HD em várias partições e mais ainda, a importância dessa metodologia. Claro, que se o usuário desejar pode ter um único ponto de montagem, / que é o diretório raiz do sistema, e as pastas normais dentro desta única partição, terá o sistema funcionando sem nenhum contratempo.

Vejamos agora a hierarquia dos diretórios do sistema, para que você possa entender de forma básica a distribuição da árvore de diretórios do FreeBSD.

/ - É o diretório inicial do sistema ou diretório raiz. É neste ponto de montagem que se baseia a montagem de todos os outros diretórios.

/bin/ - Este diretório possui todos os programas necessários para o sistema operar de forma mono e multi-usuário.

/boot/ - Todos os arquivos e programas para inicialização do sistema ficam neste diretório. Para entendermos melhor este processo, vejamos os sub-diretórios existentes:

defaults/ - Os arquivos de configuração padrão para o carregamento do sistema estão dentro deste diretório . O arquivo loader.conf possui as configurações padrão para o carregamento do sistema.

kernel/ - O kernel do sistema fica salvo neste diretório. Ao iniciar o sistema, o kernel é automaticamente carregado na memória do sistema para posteriormente ser executado. Lembrando que o kernel é um arquivo binário.

modules/ - Os módulos que podem ser carregados pelo kernel estão todos neste diretório. Exemplo de módulos: acpi e emulação linux.

/cdrom/ - Este é o ponto de montagem do CDROM. Este padrão é definido pelo configurador do sistema, o sysinstall e esta informação também está alocada em /etc/fstab.

/compat/ - Este diretório aloca os arquivos de configuração para emulação de compabilidade de aplicativos, no caso mais comum seria a emulação Linux. No caso de emulação linux, é criada uma imagem espelho de um sistema Linux com referência aos seus diretórios reais (bin, etc e outros).

/dev/ - Para entendermos melhor o diretório dev, é interessante vermos algumas teorias de engenharia do sistema FreeBSD. Todos os sistemas FreeBSD e a maioria dos sistemas Linux e Unix seguem um padrão de criar um link ao dispositivo, sendo este procecimento denominado de devfs ou seja, device file system. Então, todos os dispositivos, tais como CDROM, HD e portas seriais estão listados dentro deste diretório. Vejamos alguns exemplos:

/dev/ad0 Hard Disk ATA

/dev/acd0 CDROM ATA

/dev/cuaa0 Porta serial

fd/ - Neste diretório localiza-se o sistema descritor de arquivos. Lembrando que o sistema descritor de arquivos é responsável por gerenciar o estado de qualquer arquivo.

net/ - Todos os dispositivos referentes à rede localizam-se neste diretório, tais como interfaces ethernet, serial line (slip) e wireless.

/dist/ - Este diretório é usado basicamente pelo sysinstall, o instalador e painel de controle do FreeBSD.

/etc/ - Este diretório é um dos mais importantes do sistema, sendo composto por vários arquivos de configuração do sistema e scripts de uso geral. É interessante inclusive aprender um pouco mais sobre este diretório.

defaults/ - Este diretório contém todos os arquivos padrão do sistema, tais como o rc.conf. Serve como modelo para posterior estudo do usuário, pois nele é apresentado a maioria das variáveis existentes para cada arquivo de configuração.

gnats/ - O diretório gnats possui uma peculiaridade própria, pois serve como repositório para o envio de informações através do send-pr (Ferramenta de envio de problemas) ao time de desenvolvimento do sistema.

isdn/ - Todas as configurações básicas para o bom funcionamento do serviço de ISDN encontram-se neste diretório.

localtime/ - As informações sobre a zona horária do sistema ficam armazenadas neste diretório.

mail/ - O FreeBSD vem nativamente com o sendmail (aplicativo de envio e recebimento de e-mails) instalado e pronto para uso. Sendo assim, este diretório contém toda a parte de configuração do sendmail.

mtree/ - O utilitário mtree tem como função gerar o mapeamento dos diretórios do sistema em forma impressa ou em listagem na tela. As configurações deste aplicativo ficam armazenadas neste diretório.

namedb/ - Este diretório tem vital importância para os sistemas FreeBSD que utilizam o serviço de daemon DNS . O utilitário named , que é responsável pela transcodificação e disponibilização de nomes através de ip pela Internet, possui todos os arquivos de configuração armazenados neste diretório.

pam.d/ - O aplicativo PAM , que tem como função autenticar o usuário por dispositivos plugáveis armazena os seus arquivos de configuração neste diretório.
periodic/ - Existe um aplicativo no sistema denominado cron. A função do cron é a de realizar tarefas em tempo pré-programado pelo usuário ou por alguma aplicação. Justamente neste diretório estão os scripts para serem rodados pelo cron.

ppp/ - Este diretório guardas as informações para o bom funcionamento do utilitário ppp , que é responsável por entre outras tarefas, permitir que o usuário conecte-se na internet tanto via modem serial como via interface ethernet.

ssl/ - Os arquivos de configuração do OpenSSL, aplicativo de encriptação de dados, ficam contidos neste diretório.

/lib/ - Este diretório armazena todas as bibliotecas necessárias para o funcionamento dos mais diversos aplicativos, principalmente para os existentes em /bin e /sbin.

geom/ - As bibliotecas para o funcionamento do aplicativo geom (responsável por unir vários discos formando uma única partição)

/libexec/ - Este diretório armazena todas as bibliotecas necessárias para o funcionamento dos mais diversos aplicativos, principalmente para os existentes em /bin e /sbin.

/mnt/ - Este diretório tem a finalidade de servir de ponto de montagem temporária para discos rígidos, CDROM e outros dispositivos.

proc/ - Este diretório armazena todas as informações referentes ao sistema de arquivos de processos.

/root/ - Diretório do usuário root (administrador do sistema)

/bin/ - Os programas de administração e manutenção do sistema encontram-se em grande parte neste diretório.

/stand/ - Este diretório é usado para o ambiente sysinstall .

/tmp/ - O diretório tmp armazena informações temporárias, tais como logs de aplicações que não requerem armazenamento , bem como outros arquivos.

/usr/ - O usuário pode facilmente localizar dentro do diretório usr, a maioria dos utilitários e aplicações para seu uso.

/var/ - Os arquivos de logs podem encontram-se normalmente neste diretório, bem como arquivos de spool (impressão), lista de pacotes e outros.

3.4. USUÁRIOS, GRUPOS E PERMISSÕES

Todos os sistemas operacionais modernos podem ser utilizados por diversos usuários ao mesmo tempo, sem que seus arquivos e permissões se misturem. No FreeBSD não é diferente.

3.4.1. Usuários

Para começarmos a entender melhor o gerenciamento de usuários, vamos explicar do inicio. Existem três tipos de contas: o super usuário, usuários do sistema, e usuários. A conta “super usuário”, normalmente chamada de root, é usada para administrar o sistema sem limitação de privilégios. Os “usuários do sistema” executam serviços. Finalmente, contas de “usuários” são usadas por pessoas, que poderão logar-se ao sistema, ler e-mail, e assim por diante.

Todas as contas em um sistema FreeBSD tem determinadas informações associada à mesma, de forma à identifica-la.

- Nome do usuário

O nome do usuário será digitado no terminal de login. Os nomes de usuários devem ser únicos em um computador, não podendo haver dois usuários com o mesmo nome. Há uma série de regras para criar nomes válidos de usuários, documentadas no manual do comando passwd(5).

- Senha

Cada conta tem uma senha associada com ela. A Senha pode ser vazia, neste caso nenhuma senha será solicitada para acessar o sistema. Isso normalmente é uma ideia muito ruim.

- ID do Usuário (UID)

O UID é um número que varia de 0 a 65536 usado para identificar de forma única os usuários no sistema. Internamento, o FreeBSD usa o UID para identificar usuários. Qualquer comando do FreeBSD será liberado para um usuário específico convertendo seu UID depois de utilizar o comando. Isso significa que você pode ter contas com diferentes nomes de usuários, mas com o mesmo UID. Ambos, para o FreeBSD são um só usuário. É improvável que você tenha necessidade de fazer isso um dia.

- ID de Grupo (GID)

O GID é um número que varia de 0 a 65536, usado para identificar de forma única o grupo primário ao qual o usuário pertence. Os grupos são um mecanismo para controlar o acesso aos recursos baseados em GID ao invés de UID. Isto pode reduzir significativamente o tamanho de arquivos de configurações. Um usuário pode também pertencer a mais de um grupo.

- Classe de Login

Classe de login são uma extensão para o mecanismo de grupos que fornecem flexibilidade adicional para adaptar o sistema a diferentes usuários.

- Tempo de alteração da senha

Por padrão o FreeBSD não força os usuários a trocar as suas senhas periodicamente. Você pode fazer isto para cada usuário, forçando alguns ou todos os usuários terem que mudar suas senhas após um determinado período de tempo.

- Tempo de expiração da conta

Por padrão o FreeBSD não expira as contas. Se você estiver criando uma conta que você sabe que tem um tempo limitado de uso, por exemplo, em uma escola onde você tem contas para os estudantes, você pode especificar quando a conta deve expirar. Após a expiração deste tempo o cliente não poderá logar-se no sistema, embora diretórios e arquivos do usuário em questão, permanecerão intocados.

- Nome completo dos usuários

O nome único do usuário identifica a conta do mesmo internamente para o FreeBSD, mas não reflete necessariamente o nome real do usuário. Esta informação pode, opcionalmente, ser associada com a conta.

- Diretório Home

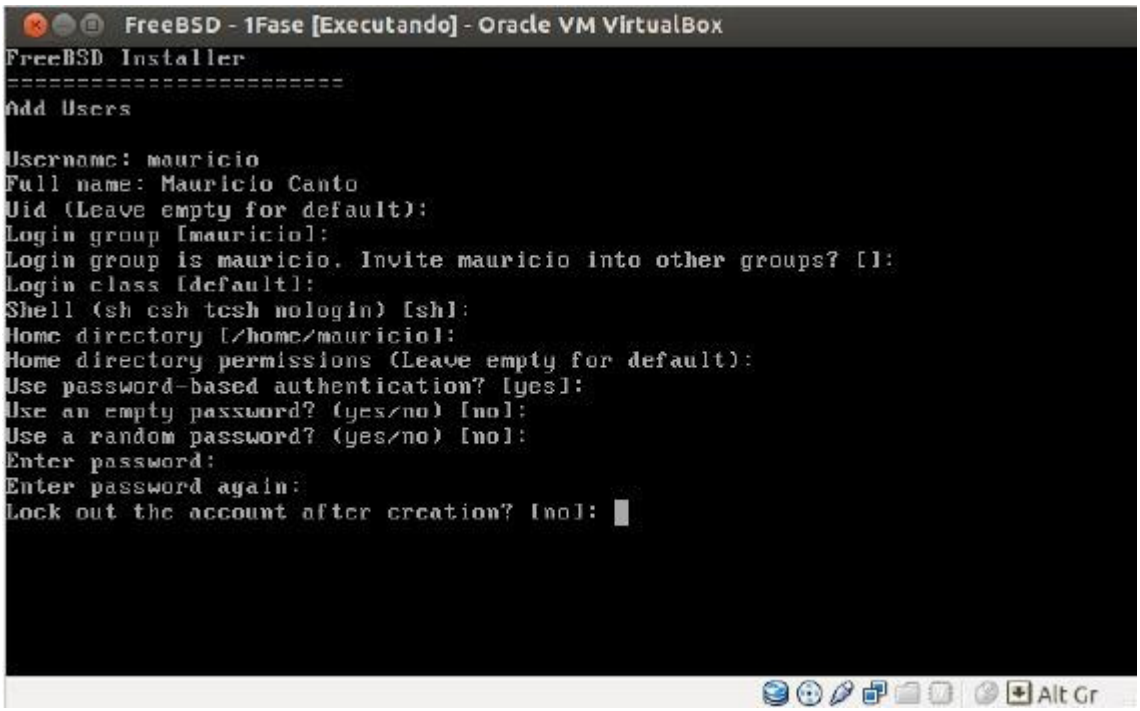
O Diretório Home é o caminho completo para o diretório do sistema em que o usuário logará. Uma convenção comum indica que todos diretórios home dos usuários devam ficar sob /home/usuário. O usuário armazenaria seus arquivos pessoais em seu diretório home, e todos os diretórios que puder criar dentro deste.

- Shell do usuário

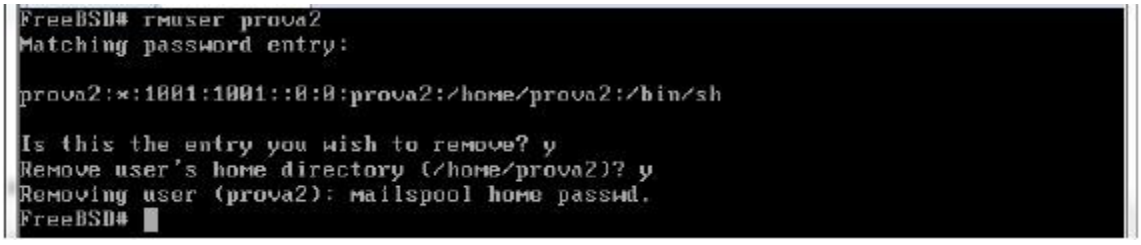
A shell fornece o ambiente padrão para o usuário interagir com o sistema. Existem muitos tipos distintos de shell, e os usuários experientes terão suas próprias preferências, que podem ser refletidas junto aos ajustes de cada conta.

Existe uma porção de comandos para gerenciar e manipular as contas de usuários no FreeBSD. Mas vamos conhecer apenas as mais utilizadas.

- adduser – comando para adicionar novos usuários. Este comando segue as mesmas etapas que vimos no inicio da instalação do FreeBSD.



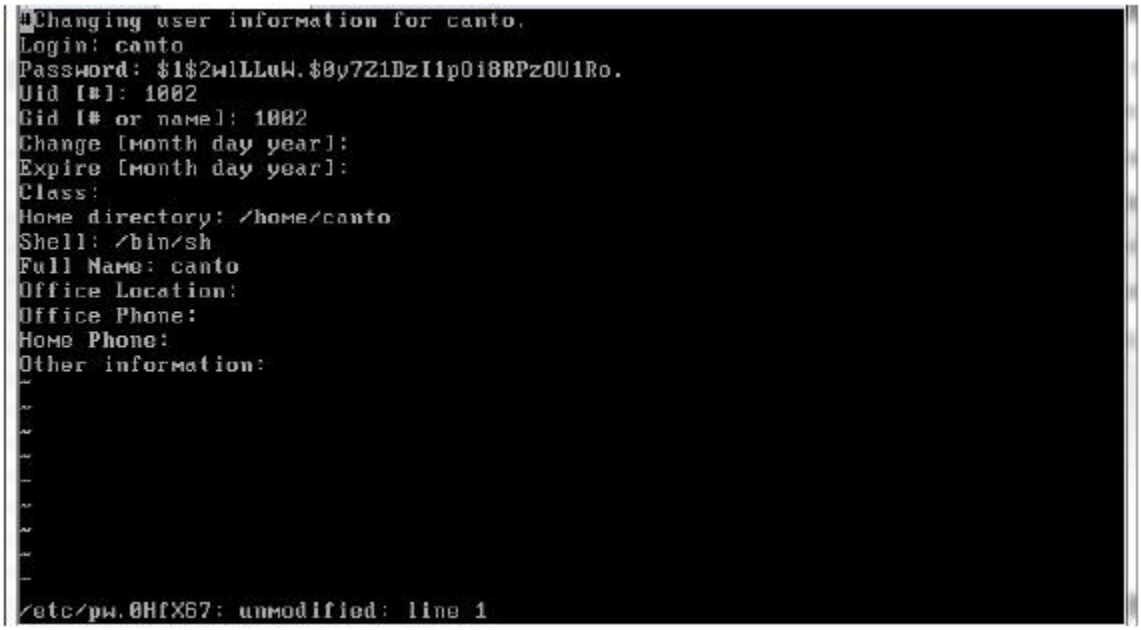
- rmuser – comando utilizado para remover todas as entradas do usuário.



Para utilizar o executar comando basta digitar o nome do usuário que desejamos excluir após o comando, conforme figura acima. Logo em seguida, virão duas perguntas. A primeira pergunta, o sistema nos mostra a entrada que fica no arquivo /etc/passwd, onde ficam todas as entradas de usuários do sistema, pedindo a

confirmação da exclusão da mesma. A segunda pergunta é para confirmar a exclusão do diretório home do usuário. É possível excluir o usuário confirmando a primeira pergunta e deixar o diretório home do usuário. Basta colocar um não como resposta da segunda questão.

- chpass – comando que possibilita alterar informações do usuário.



Ao digitar o comando e o nome do usuário que desejamos alterar, o editor VI (fala-se “vi ai”) será aberto. Dessa forma as informações podem ser alteradas e serão aplicadas após sair do arquivo.

- passwd – um simples comando para alterar a senha do usuário.

3.4.2. Grupos

O pw é um daqueles comandos de difícil entendimento no inicio, pois o seu uso é um pouco complexo e possui uma infinidade de parâmetros que podem ser utilizados. É um comando utilitário cuja função é criar, remover, alterar e apresentar usuários e grupos. Ele funciona como um front end para os arquivos de usuários e

grupos do sistema. O pw tem um conjunto muito completo de opções e comandos que o torna apropriado para o uso em scripts shell.

- pw – comando para alterar contas de usuário com grupos.

```
FreeBSD# pw group add grupo1
FreeBSD# pw group show grupo1
grupo1:*:1003:
```

Na primeira linha adicionamos o grupo1. Já na segundo estamos mostrando os dados do grupo que acabou de ser criad.

```
FreeBSD# pw group show wheel
wheel:*:0:root
FreeBSD#
```

Aqui estamos mostrando os dados do grupo wheel, que é o grupo dos administradores da máquina. Notamos que nesta tela existe uma diferença. Neste grupo possui um usuário, no anterior não existia nenhuma informação após o ":".

Agora vamos adicionar um usuário no grupo1.

```
FreeBSD# pw groupmod grupo1 -M canto,root
FreeBSD# pw group show grupo1 -P
Group Name: grupo1             #1003
   Members: canto,root

FreeBSD#
```

Na primeira linha foram adicionados os usuários canto e root no grupo1, conforme falamos. Já na segunda linha, utilizamos o parâmetro –P na visualização dos dados do grupo, que nos mostra as informações de maneira mais organizada.

Faça uma pesquisa mais afundo sobre as diversas formas de utilização do comando.

3.4.3. Permissões

Nos sistemas Unix todos os arquivos e diretórios são automaticamente assinalados por um usuário e um grupo que são "donos" deste arquivo/diretório, assim o sistema pode definir diferentes permissões de acesso entre usuários. Isto é muito importante, pois um sistema multi usuário precisa ter uma camada de segurança especial, imagine que o usuário1 cria um diretório com documentos de

texto importantes, o usuário2 por sua vez não sabe para que serve esta pasta e acaba apagando-a por engano, este tipo de coisa não pode acontecer em uma empresa. Desta forma foram criadas as permissões de acesso, que restringem as ações que dos usuários em arquivos e diretórios.

Para verificarmos as permissões de acesso à arquivos utilizamos o comando ls -l, que retorna uma lista com diversas colunas, devemos ter uma atenção especial com três destas colunas.

```
FreeBSD# ls -l
total 28
-rw-r--r--  2 root  wheel   793 Jan  3  2012 .cshrc
-rw-------  1 root  wheel  2484 Feb  5 22:26 .history
-rw-r--r--  1 root  wheel   151 Jan  3  2012 .k5login
-rw-r--r--  1 root  wheel   299 Jan  3  2012 .login
-rw-r--r--  2 root  wheel   256 Jan  3  2012 .profile
-rw-r--r--  1 root  wheel     0 Jan 23 22:34 arquivo1-copia
lrwxr-xr-x  1 root  wheel    18 Jan 30 20:40 arquivo1-renomeado -> arquivo1-reno
meado
drwxr-xr-x  2 root  wheel   512 Jan 30 20:40 novo
-rw-r--r--  1 root  wheel   191 Jan 30 22:10 redireciona.txt
FreeBSD#
```

Como podemos observar, as permissões de acesso são definidas por um conjunto de caracteres, no caso do arquivo1.txt as permissões de acesso são "-rw-r--r--", neste conjunto de caracteres estão definidas respectivamente as permissões de acesso do usuário dono, do grupo dono e dos outros usuários. Veja na figura abaixo como decifrar esta combinação de permissões:

Agora podemos ver claramente, os caracteres em azul na figura definem as permissões de acesso ao dono do arquivo os caracteres em vermelho, definem as permissões de acesso ao grupo dono do arquivo e os caracteres em verde definem as permissões de acesso aos outros usuários. Podemos observar também que sempre é utilizado um conjunto de três caracteres para cada permissão. Vamos agora traduzir o que os caracteres (triplets) representam:

R	Leitura, simbolizada pela letra r (de read)
W	Escrita, simbolizada pela letra w (de write)
X	Execução, simbolizada pela letra x (de execute)

É possível alterar estas permissões. Para isso utilizamos o comando chmod. Existem duas formas de trabalhar com o comando chmod. Uma delas é através dos triplets (rwx) vistos anteriormente, a segunda é através de uma tabela numérica que representa as letras da tabela acima.

No comando da imagem acima adicionamos permissões de escrita e execução para o grupo. Mas como isso funciona? A forma como os parâmetros são passados é do tipo usuário + permissão. Veja na tabela abaixo que tipos de usuários podem ser utilizados:

U	Usuário dono
G	Grupo dono
O	Outros usuários
A	Todos os usuários

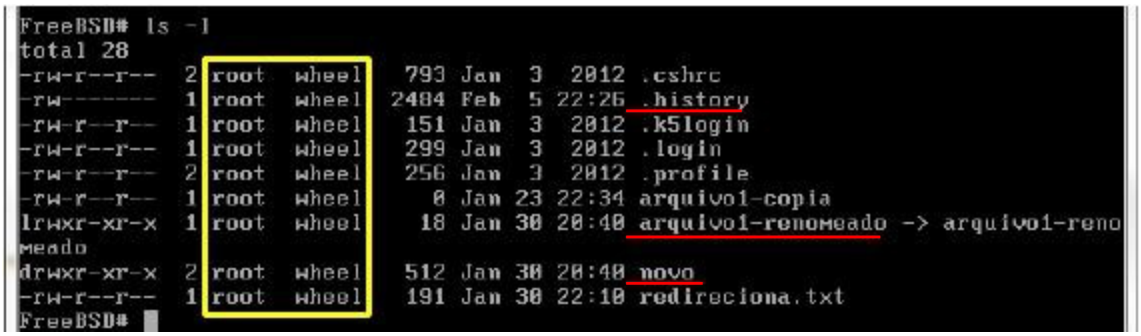
Já definição de permissões pelo modo numérico funciona de maneira diferente. De uma única vez definimos as permissões do usuário dono, do grupo dono e dos outros usuários através de um número passado como parâmetro, a tabela de números esta abaixo:

7	rwx
6	rw-
5	r-x
4	r--
3	-wx
2	-w-
1	--x
0	---

Conforme visto na tabela acima, vejamos o funcionamento do comando.

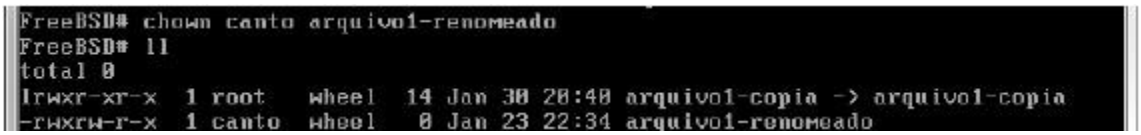
Outras informações que podemos obter é o tipo da informação que estamos listando, ou seja, o primeiro caractere que vem antes das permissões é que define isto. Se for um "-" é um arquivo, já o "d" significa que ali é um diretório e temos também o "l" que identificamos como link simbólico.

Na imagem abaixo podemos ver que o .history é um arquivo, novo é um diretório e arquivo1-renomeado é um link simbólico.



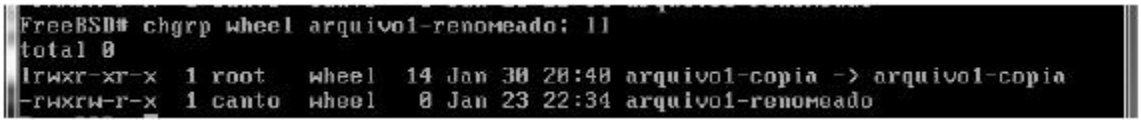
Mas uma listagem dessa pode nos mostrar ainda mais detalhes importantes para a administração do S.O. As colunas em amarelo na imagem anterior nos mostram que são os proprietários e grupos aos quais aquelas informações pertencem. Que neste o "dono" é o usuário root e o seu grupo wheel.

Assim como podemos alterar as permissões dos arquivos e diretórios, conseguimos alterar também o dono e o grupo dos mesmos. Para essas alterações utilizamos os comandos chown e chgrp.



Como visto na imagem acima, o comando é bem simples de ser utilizado. Mas com ele também podemos alterar o grupo.

Já o comando chgrp só permite que alteremos o grupo dono.



3.5. INSTALANDO PACOTES

O FreeBSD possui centenas de programas disponíveis gratuitamente para utilizarmos. Estes programas se apresentam sempre de duas formas, uma pré-compilada e outra como código fonte para que você o compile.

3.5.1. Pacotes pré-compilados

Os aplicativos pré-compilados se encontram no diretorio "packages" e devem ser instalados usando o comando pkg_add. Estes aplicativos são geralmente instalados no diretório /usr/local/bin. O uso de versões pré-compiladas traz a vantagem de ser pratica de se instalar, mas muitas vezes não traz disponível todos os recursos do aplicativo e suas dependências, como no caso do apache. O que nos obriga a instalar o aplicativo através dos ports, para poder customiza-lo na recompilação.

A utilização dos comandos pkg são simples.

- pkg_add [nome_do_arquivo].tgz
- pkg_delete nome_do_aplicativo
- pkg_info

3.5.2. Aplicativos através do “ports”

O ports é uma coleção de aplicativos que pode ser configurado durante o processo de instalação do S.O., porém existem várias formas de instalá-lo manualmente. Antes que você possa instalar um port, você primeiro precisa obter a

coleção de ports, a qual é basicamente um conjunto de arquivos Makefiles, patches, e arquivos de descrição colocados sob /usr/ports.

A lista de aplicativos disponíveis para FreeBSD cresce o tempo todo. Felizmente, existem diversas formas de procurar o que você necessita:

O web site do projeto FreeBSD mantém uma lista atualizada de todos os aplicativos disponíveis e na qual você pode executar buscas, em http://www.FreeBSD.org/ports/.

Para atualizar a sua coleção de ports você deverá executar o seguinte comando (se for a primeira atualização a ser feita após instalação do FreeBSD):

- # portsnap fetch extract

Ele irá atualizar toda a sua coleção de ports do seu FreeBSD. Quando for atualizar a coleção de ports nas próximas vezes execute somente o comando abaixo:

- # portsnap fetch update

Pronto já temos os ports atualizados o próximo passo agora é finalmente instala-lo, como já deve ser do conhecimento de todos os ports ficam localizados no seguinte caminho abaixo em seu FreeBSD:

- # cd /usr/ports

Encontrado o seu ports pelo nome é somente executar o seguinte comando dentro da pasta do aplicativo:

- # make install clean

Vamos instalar o apache 2.2 então:

- # cd /usr/ports/www/apache22/ && make install clean

Bem, se você utilizar proxy em sua rede talvez seja necessário passar isto para o fecth o aplicativo que baixar o programa para o ports realizar a instalação, para setar o proxy faça o seguinte procedimento:

- # vi /etc/make.conf

Insira a linha > FETCH_ENV = HTTP_PROXY=http://user:passwd@proxy.name:80

Para listar os detalhes dos pacotes instalados através do pkg e ports no FreeBSD:

- # pkg_info

Acesse o diretório da coleção de ports que você deseja desinstalar, por exemplo, para desinstalar o apache22:

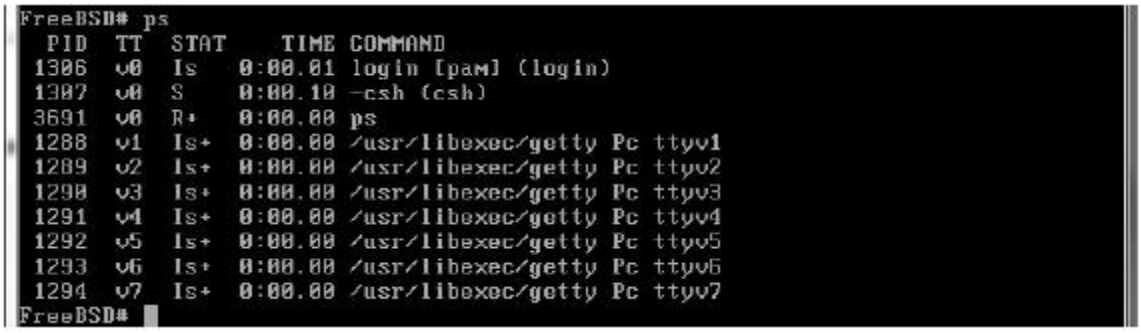
- # cd /usr/ports/www/apache22 && make deinstall

E depois limpe o "lixo" deixado pela instalação do pacote apache22, ainda na pasta execute o comando abaixo:

- # make clean

3.6. CONTROLE DE PROCESSOS

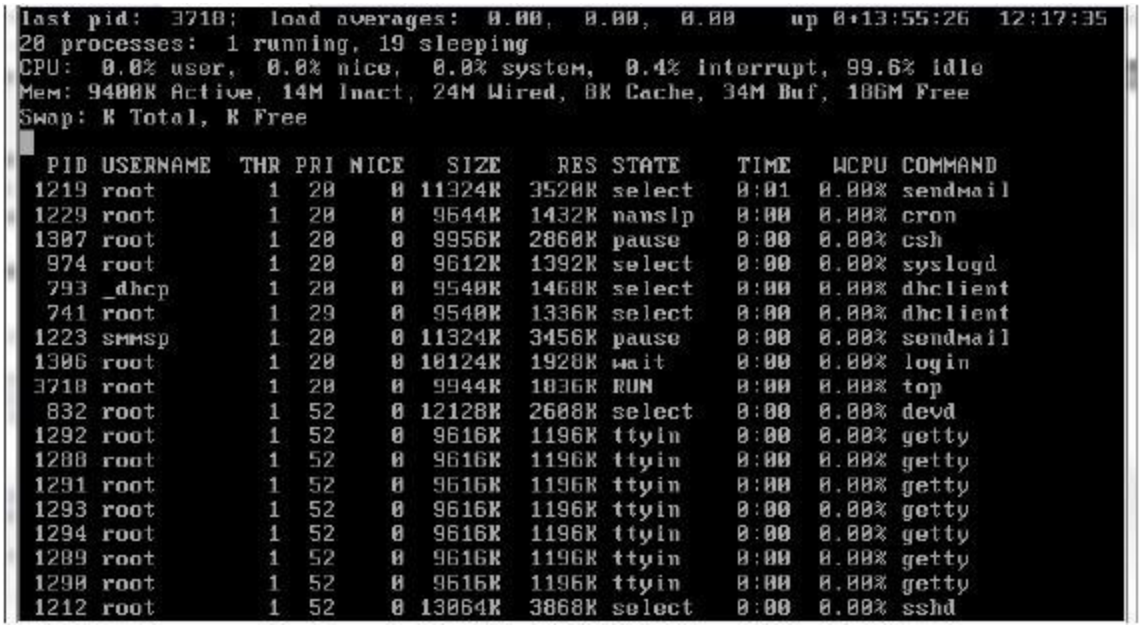
Para listarmos os processos no Linux utilizamos o comando ps, assim podemos obter os pids de processos e saber quais e quantos processos estão sendo executados no momento, para vermos os processos sendo executados no bash atual utilizamos o comando ps sem parâmetros exemplo:



PID	É a identificação do processo.
TT	Posta em que terminal o programa esta sendo executado.
STAT	Apresenta o estado atual do programa.
TIME	Indica a quantidade de tempo que o programa está em execução na CPU.
COMMAND	É a linha de comando utilizada para executar o programa em questão.

O comando ps aceita diversos parâmetros. É um comando muito utilizado com o grep para trazer os processos que estamos buscando.

Já o comando top traz uma saída muito similar com a do comando ps. Porém ele nos traz algumas informações mais importantes como a carga de uso do servidor.



A saída está dividida em duas seções. O cabeçalho (as primeiras cinco linhas) apresenta o PID do último processo executado, a média de carga do sistema (que mede o quanto o sistema está ocupado), e o tempo de atividade ininterrupta do sistema (tempo desde a última inicialização - o uptime). As outras informações do cabeçalho indicam quantos processos (20 neste caso) estão ativos, quando de memória e swap foi utilizado, e quanto tempo o sistema está gastando em estados diferentes de uso da CPU.

Abaixo está uma série de colunas com informações similares à saída do ps. Como antes, você pode ver o PID, o usuário, o tempo de CPU utilizado, e o comando executado. top também mostra por padrão a quantidade de memória utilizada pelo processo. Essa informação é dividida em dois grupos, uma para o tamanho total de memória, e um para o tamanho residente -- tamanho total indica quanta memória a aplicação já precisou, e o tamanho residente é a quantidade em uso no momento.

Para finalizarmos um aplicativo ou um programa travado utilizamos o comando kill <pid>, desta forma o programa vai ser fechado:

- # kill 793

3.7. SHELLS

No FreeBSD, uma grande parte do trabalho do dia a dia é feito por interface de linha de comando, chamada de interpretador de comandos, ou shell. A principal função de uma shell é receber comandos por um canal de entrada e executá-los. Uma série de shells ainda possuem funções embutidas, o que ajuda a facilitar a realização de tarefas do dia-a-dia, como gerenciamento de arquivos, edição de linha de comando, macros de comandos e variáveis de ambiente. O FreeBSD vem com uma série de interpretadores de comandos, como o sh, chamado de Bourne Shell, e tcsh. A "C-shell" melhorada. Muitos outros interpretadores de comandos estão disponíveis na coleção de ports do FreeBSD, como o zsh e o bash.

Que shell utilizar? Trata-se meramente de uma questão de gosto. Se você é um programador C, se sentirá mais confortável com um interpretador de comandos similar à linguagem C, como o tcsh. Se você teve suas origens no Linux, ou é novo ao UNIX®, tenta experimentar o bash. O ponto é que cada interpretador de comandos possui propriedades únicas, que pode ou não funcionar corretamente em seu ambiente de trabalho, e também, outro ponto importante é sua liberdade de escolha para decidir qual e quando interpretador utilizar.

Uma das características mais comuns em interpretadores de comandos é completar nome de arquivos. Ao digitar as primeiras letras de um comando ou de um arquivo, você normalmente pode fazer com que a shell automaticamente complete o restante do comando ou arquivo, pressionando a tecla "Tab" no teclado. Aqui está um exemplo. Suponha que você tenha dois arquivos chamados foobar e foo.bar. Você quer apagar o foo.bar. Então o que você faria no teclado é: rm fo[Tab].[Tab].

O interpretador de comandos apresentaria como resultado algo similar à rm foo[BEEP].bar.

O [BEEP] é o sino do console, que indica que o interpretador de comandos não foi capaz de completar totalmente o nome do arquivo porque existe mais do que uma opção que coincide. Ambos, foobar e foo.bar começam com fo, mas a shell pode completar o nome do arquivo até foo. Se neste momento você digitar ., e pressionar

Tab mais uma vez, a shell deve ser capaz de completar o restante do nome do arquivo para você.

Outra funcionalidade de um interpretador de comandos é o uso de variáveis de ambiente. Variáveis de ambiente são duplas variáveis de valores armazenados no espaço de ambiente da shell. Esse espaço de ambiente pode ser lido por qualquer programa chamado pelo interpretador, de forma que contenha uma série de configurações interpretadas pelo mesmo. Segue aqui uma lista de variáveis de ambiente comuns, e seus respectivos significados:

Variável	Descrição
USER	Nome do usuário logado no momento.
PATH	Lista de diretórios, divididos por vírgula, indicando onde deve-se procurar por binários.
DISPLAY	Nome de rede do terminal gráfico do X11 onde o mesmo deve se conectar, caso esteja disponível.
SHELL	O interpretador de comandos atual.
TERM	Nome do terminal do usuário. É usado para determinar as características do terminal.
TERMCAP	Entrada com a base de dados de códigos de terminal, que realizam várias funções do console.
OSTYPE	A espécie do sistema operacional em questão, por exemplo, FreeBSD.
MACHTYPE	A arquitetura de CPU onde o sistema está sendo executado.
EDITOR	Editor de texto padrão para o usuário.
PAGER	O paginador de texto padrão para o usuário.
MANPATH	Lista de diretórios separadaos por vírgulas que indicam onde as páginas de manuais devem ser buscadas.

Ajustar as variáveis de ambiente é tarefa que se difere de algumas formas entre as várias shells disponíveis. Por exemplo, nas shells baseadas na linguagem C, como o tcsh e o csh, você usaria o setenv para justar as variáveis de ambiente. Sobre um interpretador estilo Bourne shell como o próprio sh e o conhecido bash, você usaria o export para definir a variável de ambiente. Por exemplo, para definir ou

modificar o valor da variável de ambiente EDITOR, sob csh ou tcsh, para definir a variável EDITOR com o valor /usr/local/bin/emacs, você faria algo como:

% setenv EDITOR /usr/local/bin/emacs

Enquanto sob uma Bourne shell seria:

% export EDITOR="/usr/local/bin/emacs"

Ainda é possível fazer a maioria dos interpretadores expandirem a variável, simplesmente adicionado o carácter $ na frente da própria variável. Por exemplo, echo $TERM imprimiria todo o conteúdo da variável $TERM, por que o interpretador interpreta a variável $TERM como seu valor, e o passa para o echo.

Os interpretadores de comandos cuidam ainda de uma série de caracteres especiais, chamados de meta-caracteres, como representações especiais de dados. O caractere mais comum é o *, que representa qualquer conjunto de caracteres em um nome de arquivo. Os meta-caracteres especiais podem ser usados como substituição de arquivos. Por exemplo, o comando echo * tem resultado similar à saída de um ls, pois o interpretador de comandos pega todos os arquivos que batem com * e os coloca na linha de comando de forma que o echo possa enxerga-los.

Para evitar que a shell interprete estes caracteres especiais, pode-se utiliza-los como exclusão, inserindo uma barra invertida (\) na frente deles. echo $TERM imprime em que terminal você está, enquanto echo \$TERM imprime $TERM exatamente.

A forma mais fácil de modificar seu interpretador de comandos é usar o comando chsh. Executar o chsh te colocará no editor definido na variável de ambiente EDITOR; se esta variável não estiver definida, você será colocado no vi. Modifique a linha ``Shell:'' de acordo com suas necessidades.

Você pode ainda usar o chsh com a opção -s; assim você pode ajustar sua shell sem entrar no editor. Por exemplo, se a intenção era modificar seu interpretador para usar o bash, o truque pode ser feito da seguinte forma:

% chsh -s /usr/local/bin/bash

Utilizar o chsh sem parâmetros e editar a shell também funcionaria.

Nota: O interpretador de comandos que você deseja utilizar deve estar listado no arquivo /etc/shells. Se você instalou o interpretador a partir da Coleção de ports, isso já foi feito. Se você instalou na mão, você mesmo deve fazê-lo.

Por exemplo, se você instalou o bash manualmente e o colocou no /usr/local/bin, você vai querer fazer o seguinte:

```
# echo "/usr/local/bin/bash" >> /etc/shells
```

Em seguida, execute novamente o chsh.

4 MISCELÂNEA

Existem uma infinidade de distribuições e customizações de distribuições na internet. Projetos consolidados, com boa documentação de refêrencia. Vale ressaltar que o espírito de liberdade do software livre (conforme vimos no tópico 1.1.1), promove a melhoria constante dos SO de maneira colaborativa e descentralizada – A todo momento várias pessoas trabalham sem ganhar nada por isso, na melhoria e customização de distribuições, ou criando sua própria. Será relacionado aqui alguns links úteis referentes aos assuntos já vistos, e também uma breve descrição de outras distribuições.

4.1 CONHECENDO OUTRAS DISTRIBUIÇÕES

Como citado anteriormente uma gama de distribuições já existentes e outras em crescimento, estão a nossa disposição para uso. A melhor distribuição é muito relativo e depende da familiaridade de cada usuário e os fins de utilização. Vamos conhecer brevemente as distribuições principais da família BSD.

NetBSD – Deriva do Unix BSD e é baseado em software livre e escrito em mutiplataforma.

OpenBSD – Deriva do NetBSD, é completamente dedicado a servidores. Tem a fama de ser o sistema mais seguro e estável do mundo. Infelizmente existe pouca documentação em português.

4.2 LINKS ÚTEIS

Afim de organização e facilitação dos estudos será relacionada neste tópico alguns links importantes sobre os temas abordados anteriormente.

NetBSD – Guia do Usuário em Português
http://web.mclink.it/MG2508/nbsdbra/netbsd.html

Site principal com histórico, tutoriais, ports e etc (em inglês)- http://www.netbsd.org/

Download (em formato ISO – versão corrente 6.1)- http://www.netbsd.org/releases/

OpenBSD – Manual do usuário em inglês.
http://www.openbsd.org/cgi-bin/man.cgi

Site principal (em inglês)- http://www.openbsd.org/

Download - http://www.openbsd.org/ftp.html

FreeBSD – Site principal (em inglês)- http://www.freebsd.org/

Download - http://www.freebsd.org/where.html

Handbook – guia completo de uso com descrição de comandose etc- em português. http://www.vivaolinux.com.br/publico/FreeBSD.Handbook.pdf