

# **Gestão de Sistemas Operacionais II**



## **Apostila orientada as Bases Tecnológicas**

- 1 - Software livre: o histórico do Linux.
- 2 - Instalação e desinstalação do Linux.
- 3 - Armazenamento de dados; sistemas de arquivos.
- 4 - A Shell
- 5 - Trabalhando com arquivos e pastas.
- 6 - Administração de contas de usuários e grupos.
- 7 - Controle e segurança de arquivo.
- 8 - O ambiente gráfico e seus aplicativos.
- 9 - Administração de redes em software livre.
- 10 - Aplicativos de apoio (S.O. de apoio: Linux).

Este material possui trechos do livro: (Morimoto C. E, Linux Guia Pratico, 1ª Edição, São Paulo: GDH Press e Sul Editores. 2009. ISBN: 978-85-99593-15-8) e outros materiais encontrados na internet. Um apanhado selecionado de tutoriais da Web.

Professor: Wellington Martins

## 1. Software livre: o histórico do Linux

Segundo Morimoto, Carlos E. O sistema operacional é o responsável por ativar todos os periféricos e criar o ambiente sobre o qual todos os outros programas rodam.

O domínio da Microsoft na área de sistemas operacionais começou em 1981, com o lançamento do primeiro PC e da primeira versão do MS-DOS. Embora não tivesse nada de especial com relação a outros sistemas da época, o DOS cresceu em popularidade junto com os PCs, seguido pelas diversas versões do Windows. Apesar disso, a Microsoft é uma página recente na história da informática. Enquanto o MS-DOS ainda dava seus primeiros passos, o Unix já era um sistema maduro, usado na maioria dos computadores de grande porte e em estações de trabalho. A história do Unix começou em 1969, na frente de um computador igual a este:



Tudo começou em 1991, quando Linus Torvalds começou a trabalhar no desenvolvimento de um sistema Unix para rodar em seu 386. Na época, o único sistema similar era o Minix, um sistema operacional para uso acadêmico, que era bastante limitado. No início, Linus usava o Minix para rodar o editor, compiladores e outras ferramentas de desenvolvimento que utilizava para desenvolver o kernel Linux, mas, a partir de um certo ponto, ele passou a usar o próprio Linux. Ou seja, depois de um breve período de encubação dentro do Minix, o Linux passou a ser desenvolvido dentro do próprio Linux.

De início, o kernel Linux era um projeto muito pequeno, o hobby de um único programador. Entretanto, ele tinha uma grande vantagem em relação aos sistemas UNIX que o precederam: o simples fato de ser disponibilizado sob a licença GPL. Isso permitiu que outros programadores adotassem o projeto, passando a contribuir com melhorias e correções.

### 1.1 A licença GLP.

A licença GPL, tão comentada, mas ao mesmo tempo tão mal compreendida, pode ser resumida em quatro direitos básicos e uma obrigação:

1- Aplicativos disponibilizados sob a GPL podem ser usados por qualquer um e para

qualquer fim, sem limitações. Mesmo que eventualmente os criadores mudem de ideia e resolvam passar a distribuir novas versões do programa sob outra licença, as versões que foram distribuídas sob a GPL continuam disponíveis, o que permite que outros desenvolvedores criem uma derivação e continuem o desenvolvimento. Isso traz uma boa dose de segurança para quem usa o aplicativo, já que reduz a chance de ele ser descontinuado e ficar indisponível. Enquanto houver um volume considerável de usuários interessados no aplicativo, é bem provável que o desenvolvimento continue de uma forma ou de outra.

2- Direito de tirar cópias do programa, distribuí-las ou até mesmo vendê-las a quem tiver interesse. Existe a possibilidade de ganhar algum dinheiro vendendo CDs gravados, por exemplo, mas como todo mundo pode fazer a mesma coisa, é preciso vender por um preço relativamente baixo, cobrando pelo trabalho de gravação e não pelo software em si, que está largamente disponível.

Isso faz com que a forma mais eficiente de ganhar dinheiro com os softwares seja prestar suporte e vender serviços de personalização e não com a venda direta, como no caso dos softwares comerciais. Para o cliente, acaba sendo vantajoso, pois o custo de implantação será o gasto com a consultoria e treinamentos, enquanto ao implantar um software comercial qualquer, ele gastaria também com as licenças de uso.

3- Direito de ter acesso ao código fonte do programa, fazer alterações e redistribuí-las. Para um programador este é o principal atrativo, já que permite criar novos projetos usando como base o código fonte de programas já existentes (ao invés de ter sempre que começar do zero), sem falar na grande oportunidade de aprendizado que examinar o código fonte de outros programas propicia.

4- Direito (e ao mesmo tempo a obrigação) de redistribuir as modificações feitas. Este é o ponto onde existem mais mal-entendidos. Se você desenvolve um software por hobby, ou para usá-lo internamente na sua empresa, e não possui interesse em explorá-lo comercialmente, você pode simplesmente divulgar o código fonte para todo mundo, o que é o caminho mais lógico se você pretende atrair outros interessados em ajudá-lo no desenvolvimento. Mas, caso você pretenda receber pelo seu trabalho de desenvolvimento, existem duas opções:

a) Você pode distribuir o software livremente para aumentar a base de usuários e ganhar vendendo suporte, treinamentos e personalizações.

b) Você só é obrigado a distribuir o código fonte a quem obtém o software, de forma que você pode trabalhar batendo de porta em porta, vendendo o software para alguns clientes específicos e fornecendo o código fonte apenas para eles. Não existe nada de errado com este modelo, mas você perde a possibilidade de ter contribuições de outros desenvolvedores, o que pode ser ruim em longo prazo.

Os softwares distribuídos sob a GPL também não "contaminam" softwares comerciais ou de outras licenças no caso de distribuição conjunta. Por exemplo, uma revista pode distribuir alguns softwares GPL no meio de um monte de aplicativos proprietários na mesma edição. Os softwares GPL continuam sendo GPL, com todas regras que vimos acima, enquanto os softwares proprietários continuam sendo fechados. A revista deve incluir o código fonte dos aplicativos GPL (ou pelo menos a informação de como obtê-los via Internet), mas, naturalmente, não precisa fazer o mesmo com os outros aplicativos incluídos no CD.

Você pode também usar algum software GPL em conjunto com o seu aplicativo

comercial, desenvolvendo um aplicativo qualquer que utiliza o PostgreSQL (um servidor de banco de dados), por exemplo. O PostgreSQL continua sendo GPL e o seu aplicativo continua sendo fechado; qualquer um pode usar e tirar cópias do PostgreSQL, mas você controla a distribuição do seu aplicativo. Uma coisa não interfere com a outra.

Ou seja, muito embora alguns vejam a GPL como algum tipo de licença comunista, que diz que todos os programadores devem fazer voto de miséria e passar a trabalhar de graça em nome do bem comum, ela é na verdade apenas uma licença que estimula a colaboração e o reaproveitamento de softwares e componentes, e que vem nos trazendo diversas mudanças positivas. De certa forma, podemos dizer que a GPL é uma licença até bastante capitalista (no bom sentido), pois estimula a concorrência entre projetos e empresas e dificulta a criação de monopólios, que são ruins para o sistema econômico.

## 1.2 A árvore genealógica das distribuições

No começo, instalar o Linux era uma tarefa ingrata. Tudo o que existia era o código-fonte do kernel, que precisava ser compilado (usando o Minix ou outro sistema operacional) e combinado com outros utilitários e bibliotecas (que também precisavam ser compilados, um a um) para que você tivesse um sistema operacional funcional. Isso explica por que nos primeiros meses, após o célebre anúncio feito por Linus Torvalds em agosto de 1991, o Linux tinha apenas algumas dezenas de usuários, a maior parte deles programadores, que em maior ou menor grau participavam do desenvolvimento do sistema.

Alguém chegou a uma conclusão óbvia: por que não distribuir versões já compiladas do sistema, que pudessem ser instaladas diretamente? Surgiram então as primeiras distribuições Linux, que rapidamente passaram a ganhar novos adeptos.

Atualmente existe mais de 500 distribuições Linux, contando apenas as ativas, porém existe menos de 10 distribuições principais (**Debian**, **Red Hat/Fedora**, **Mandriva**, **Ubuntu**, **Slackware**, **Gentoo**, etc.) das quais todas as outras são derivadas.

Por mais diferente que seja a aparência e a escolha de softwares pré-instalados, as distribuições derivadas mantêm muitas das características da distribuição-mãe, de forma que se você consegue aprender a trabalhar com as distribuições principais, passa a não ter grandes problemas ao trabalhar com qualquer uma das distribuições derivadas delas.

A distribuição mais antiga ainda ativa é o **Slackware**, lançado em julho de 1993. Tem como objetivo preservar a tradição dos sistemas Unix, provendo um sistema estável, organizado, mas com poucas ferramentas automatizadas, o que te obriga a estudar e ir mais a fundo na estrutura do sistema para conseguir usar.

Muita gente usa o Slackware como ferramenta de aprendizado, encarando os problemas e deficiências como um estímulo para aprender.

Pouco depois, em novembro de 1994, foi lançado o **Red Hat**, que foi desenvolvido com o objetivo de facilitar a configuração e tornar o uso do sistema mais transparente, permitindo que ele atingisse um público mais abrangente. Apesar de sua alma comercial, todas as ferramentas desenvolvidas pela equipe do Red Hat tinham seu código aberto, o que possibilitou o surgimento de muitas outras distribuições derivadas dele, incluindo o Mandrake (França), o Conectiva (Brasil) e o SuSE (Alemanha).

Finalmente, temos o **Debian**, que é provavelmente a maior distribuição Linux não-comercial, tanto em volume de desenvolvedores quanto em número de usuários, diretos e indiretos.

O **Debian** em si é bastante cru em termos de ferramentas de configuração e por isso é mais popular em servidores do que em desktops. Entretanto, por oferecer um repositório de pacotes incrivelmente completo, o Debian é usado como base para o desenvolvimento de inúmeras outras distribuições, que combinam os pacotes dos repositórios do Debian com personalizações, scripts e componentes adicionais, de forma a atingirem nichos específicos.

Um exemplo de destaque é o **Knoppix**, cuja versão 3.0 (a primeira a ganhar notoriedade) foi lançada em julho de 2002. O Knoppix acabou se tornando um marco dentro da história do Linux por dois motivos. O primeiro é que ele foi a primeira distribuição Linux live-CD realmente utilizável, oferecendo um bom desempenho e um excelente script de autoconfiguração, que detectava o hardware da máquina durante o boot, gerando os arquivos de configuração de forma automática e entregando um sistema funcional no final do processo. Distribuições live-CD anteriores, como o DemoLinux, eram muito mais lentas, limitadas e não práticas para usar.

O segundo motivo, e talvez o mais importante, era a possibilidade de remasterizar o CD, gerando uma distribuição personalizada. Graças a isso, o Knoppix deu origem a um enorme número de novas distribuições, como o Kanotix (que deu origem ao atual Sidux), o Morphix (que, devido à sua arquitetura modular, ajudou a criar toda uma nova família de distribuições) e o **Kurumin** (Desenvolvido pelo próprio Morimoto).

Também derivado do Debian, o **Ubuntu** é provavelmente a distribuição Linux mais usada atualmente. Ele é desenvolvido pela Ubuntu Foundation, uma organização sem fins lucrativos, que por sua vez é patrocinada pela Canonical Inc., que ganha dinheiro vendendo suporte, treinamentos e customizações do Ubuntu. Esta combinação de ONG e empresa têm dado muito certo, combinando os esforços de um sem-número de voluntários e um grupo de desenvolvedores bem pagos que trabalham em tempo integral no desenvolvimento do sistema.

Ao invés do tradicional 1.0, 2.0, 3.0, etc., o Ubuntu usa um sistema de numeração das versões bastante incomum. Os releases são numerados com base no mês e ano em que são lançados e recebem um codinome. A primeira versão oficial foi o Ubuntu 4.10 (lançado em outubro de 2004), apelidado de "Warty Warthog", seguido pelo 5.04 (lançado em abril de 2005), apelidado de "Hoary Hedgehog" e pelo 5.10 (outubro de 2005), batizado de "Breezy Badger".

Os próximos foram o 6.06 (Dapper Drake), 6.10 (Edgy Eft), 7.04 (Feisty Fawn), 7.10 (Gutsy Gibbon), 8.04 (Hardy Heron), o 8.10 (Intrepid Ibex) e o 9.04 (Jaunty Jackalope).

Apesar de ser distribuído em um único CD, o Ubuntu utiliza um repositório bastante completo. Ao instalar o sistema, você tem um desktop pré-configurado, contendo um conjunto básico de aplicativos, que você pode personalizar instalando pacotes adicionais. Os repositórios do Ubuntu são construídos a partir do repositório unstable do Debian, processo no qual os pacotes recebem correções diversas e são recompilados, gerando o repositório "Universe".

O Ubuntu deu origem a diversas distribuições, como o Kubuntu (baseado no KDE), o Xubuntu (baseado no XFCE) e assim por diante, que compartilham o mesmo repositório, mas são baseadas em conjuntos diferentes de pacotes.

Está disponível também o "Server Edition", uma versão destinada a servidores, que é baseada no mesmo repositório, mas instala apenas os componentes básicos do sistema, criando uma instalação enxuta, à qual você pode adicionar apenas os serviços e os

componentes desejados.

Em resumo, podemos classificar as distribuições Linux em **três grandes famílias**: as derivadas do **Red Hat**, como o Fedora e o Mandriva, as derivadas do **Debian**, como o Ubuntu e o Kubuntu e as derivadas do **Slackware**, como o Slax.

Você pode ver uma tabela mais completa com as origens de cada distribuição neste link do Distrowatch: <http://distrowatch.com/dwres.php?resource=independence>.

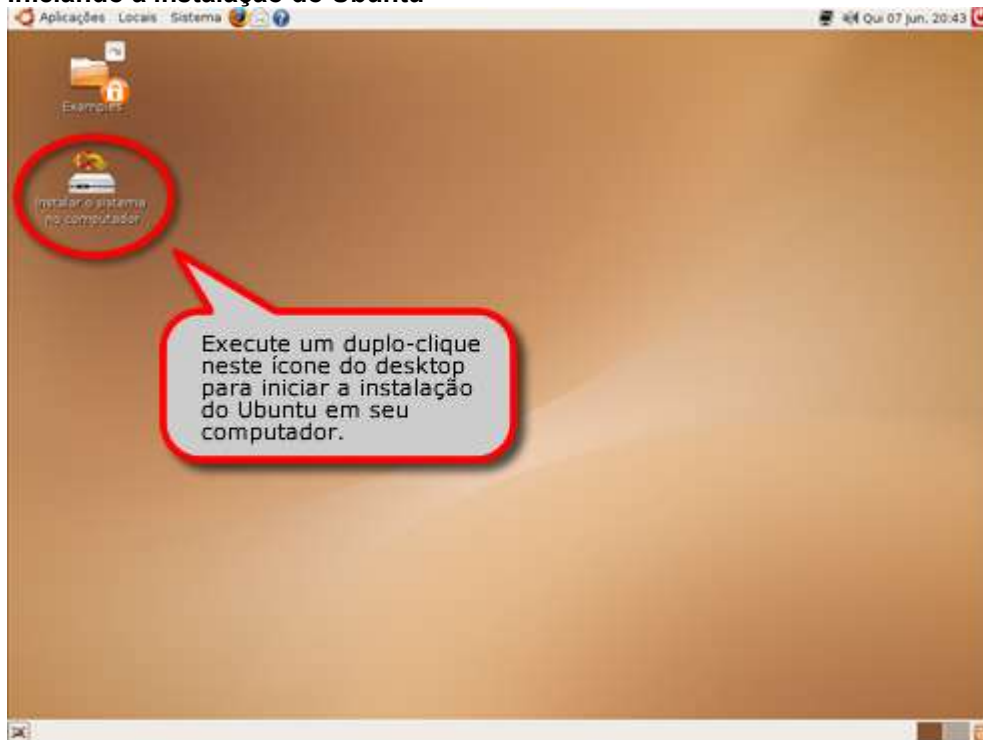
## 2. Instalação e desinstalação do Linux.

Veremos a seguir um **tutorial** simples da instalação da distribuição Linux **Ubuntu** que será realizada durante a aula. Não necessariamente com a mesma versão do sistema, ou uma mais atualizada.

### Instalando o Ubuntu a partir do Live CD

Vamos acompanhar agora um passo-a-passo de como proceder a instalação do Ubuntu GNU/Linux a partir do Desktop CD (Live CD) após sua inicialização, conforme visto anteriormente.

#### Iniciando a instalação do Ubuntu



Para realizar a instalação do sistema no Disco Rígido (HD) basta acessar a opção Instalar o Sistema no Computador presente na tela principal do sistema.

A partir desta ação, é necessário seguir os passos de acordo com o desejado, sendo que deste ponto em diante a instalação será seguida por telas visuais muito simples e intuitivas que irão guiar o usuário durante todo processo.

O sistema pergunta ao usuário se ele realmente quer continuar a instalação na linguagem já escolhida anteriormente. O idioma escolhido será o idioma padrão de todo o sistema, ou seja, se escolhermos o Português do Brasil terá toda a parte gráfica do sistema adaptada à linguagem escolhida, bem como arquivos de ajuda.



A estrutura do Ubuntu GNU/Linux é toda baseada em Live CD, permitindo ao usuário rodar o sistema sem necessariamente instalá-lo na máquina. O que o processo de instalação faz, na medida do possível é justamente jogar todos os dados do sistema em um Disco Rígido (HD).

### Selecionando o Fuso Horário



Mantida ou trocada a opção de idioma desejado, o sistema avança para a escolha do Fuso Horário. Neste ponto, é necessário escolher a zona de horário de acordo com a sua região.

Caso o sistema já tenha um link disponível de Internet, pode-se antecipadamente realizar a sincronização com algum servidor de horário caso assim desejado.

Como podemos perceber, o horário atual fica atrasado 3 horas em relação ao horário original do sistema. É necessário realizar a sincronização ou configuração manual do horário. Basta clicar então no botão Setar Hora para realizar esta alteração.

Podemos sincronizar o relógio automaticamente acessando a opção Sincronizar Agora ou ainda realizando a configuração manual acertando os parâmetros de data e hora.



Feito este procedimento, basta prosseguirmos com a instalação clicando em Avançar.

### Selecionando um Layout de teclado



Agora nos será solicitado a escolher o layout do teclado, de acordo com o dispositivo existente no computador.

A escolha pode ser testada na caixa de teste, digitando os caracteres necessários para verificação do mesmo.

Normalmente para nosso idioma, o layout de teclado escolhido deve ser o **Português Brasileiro (ABNT2)**.

Clique em Avançar para continuar com o processo de instalação.

### Preparando o Disco para instalação (Particionamento)

Muita atenção agora, pois esta é uma parte essencial para a instalação do sistema.

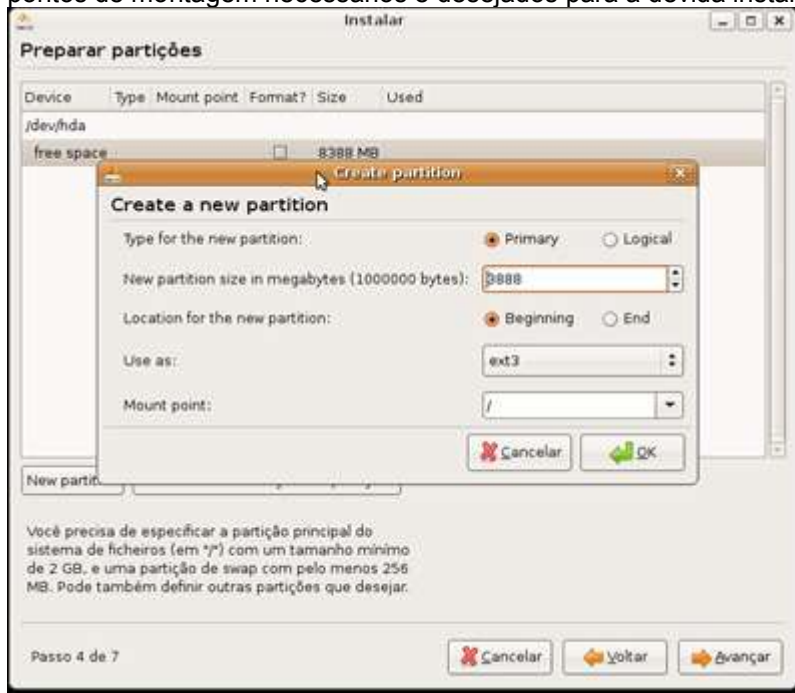


Se o computador à ser instalado o sistema não possui nenhum outro Sistema Operacional em conjunto, basta acessar a opção Assistido - Usar disco inteiro, que apaga todo o disco e o prepara automaticamente

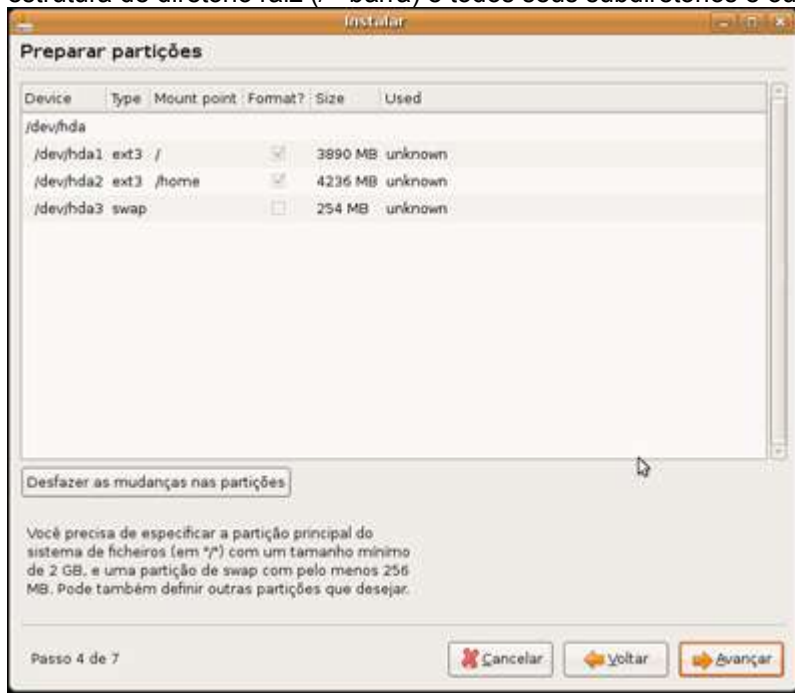


para o recebimento dos arquivos do sistema.

Caso contrário, será necessário escolher a opção de particionamento Manual, e definir todas partições e pontos de montagem necessários e desejados para a devida instalação do seu sistema.

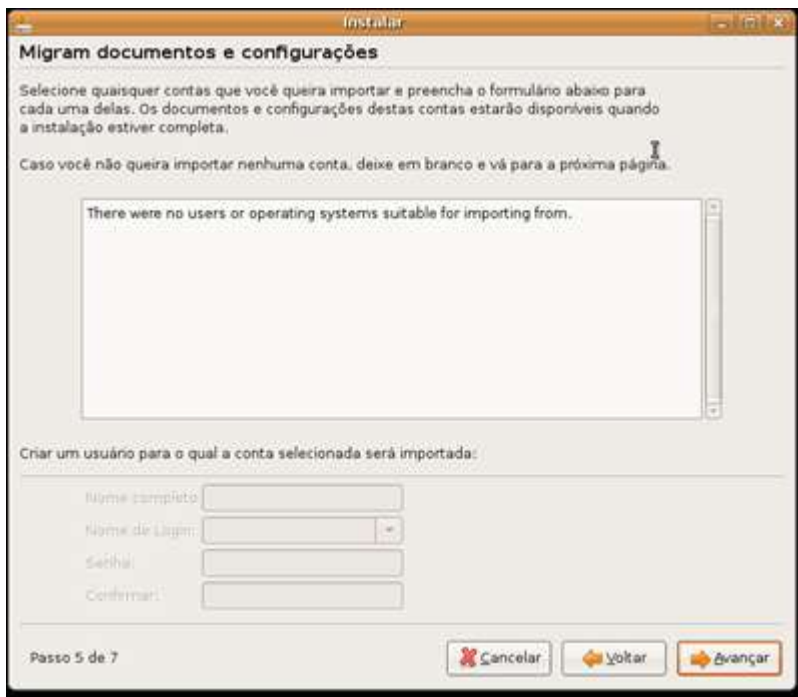


Normalmente como padrão, basta definir dois partições sendo uma para o sistema, onde será montado a estrutura de diretório raiz (/ - barra) e todos seus subdiretórios e outra como área de troca (**swap**).



Porém, sempre é aconselhável configurarmos um mínimo de três partições, sendo duas conforme visto acima e outra para montar o diretório pessoal dos usuários (**/home**). Lembre-se de sempre configurar suas partições de modo que a área de **swap** seja a última partição do disco.

## Importando configurações de outro Sistema



Caso já tenha outro sistema operacional previamente instalado em seu computador, o Ubuntu ira reconhecer o mesmo e lhe mostrar uma tela onde será possível importar do mesmo suas configurações de conta de usuário.

Selecione as opções desejadas e concluído este processo as configurações de sua conta e seus documentos pessoais ficarão disponíveis para uso.

Caso não tenha nenhum outro sistema operacional previamente instalado (conforme imagem ao lado), basta seguir adiante com processo de instalação pressionado o botão Avançar.

#### Definindo o usuário do sistema

Este passo sem dúvida nenhuma é um dos mais importantes exigidos pelo instalador do sistema. É necessário informar corretamente todos os dados exigidos, tal como nome do usuário, nome para entrada no sistema, senha do usuário e nome do computador.



É recomendável você escolher uma boa senha, com preferência em mistura de número e letras para maior segurança. Também é interessante evitar escolher senhas que contemham datas, números de telefone ou qualquer espécie que tenha ligação com a sua pessoa. Concluído esta etapa, basta clicar no botão Avançar

para dar seguimento a instalação.

### Pronto para instalar



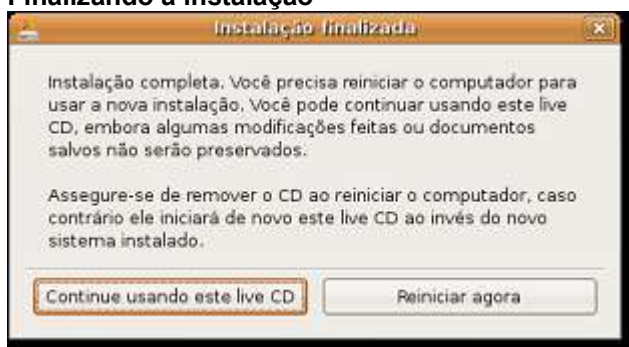
Finalizado todas etapas com sucesso, uma tela exibindo um resumo com todas informações previamente configuradas é exibida.

Confira todas informações, caso alguma coisa não esteja correta esta será a última oportunidade de retornar e corrigir antes de instalar o sistema em seu computador.

Estando tudo correto basta pressionar o botão Install. Deste ponto em diante, o Ubuntu GNU/Linux irá realizar a instalação do sistema automaticamente.

Na possibilidade de já existir uma conexão de Internet auto configurável, como DHCP habilitado na rede, o sistema verifica a versão de alguns arquivos necessários à instalação e os atualiza.

### Finalizando a instalação



Durante a instalação uma tela exibindo o status do processo será exibida.

Com o término da instalação, basta acessar a opção Reiniciar agora, para que a máquina reinicie com o sistema Ubuntu GNU/Linux de forma nativa.

Lembre-se de retirar o CD do drive para que não ocorra o boot novamente pelo mesmo.

Iniciando e logando pela instalação

Após a instalação ser feita com sucesso, finalmente podemos ter a oportunidade de entrar pela primeira vez no sistema. Basta inserir o seu usuário e respectiva senha, para utilizar o sistema.



Desfrute então da comodidade em usar um sistema operacional totalmente livre, que une beleza, praticidade e uma série de outros benefícios que só o tempo lhe mostrará.

**Créditos** (Wikifier: **Arlei** Data: **08/06/2007** Mantenedor: **Arlei** Time de Documentação do Ubuntu Brasil).

### 3. Armazenamento de dados; sistemas de arquivos.

A lista de sistemas de arquivos suportados pelo Linux é muito grande, mas discutiremos somente os mais comuns.

Atualmente, uma importante característica dos atuais sistemas de arquivos é o "journaling". Sistemas de arquivos que possuem essa característica são preferidos em detrimento aos que não possuem.

Journaling é um recurso que permite recuperar um sistema após um desastre no disco (ex.: quando um disco está sujo) em uma velocidade muito maior que nos sistemas de

arquivos sem journaling.

Segue abaixo uma breve descrição sobre os sistemas de arquivos mais comuns disponíveis para o Linux:

## **Ext2**

O sistema de arquivos ext2 é conhecido como "Second Extended FileSystem". Foi desenvolvido para ser mais "eficiente" que o sistema de arquivos "Minix", seu antecessor.

O Minix era muito utilizado nas primeiras versões do Linux, e foi utilizado por muitos anos.

O sistema de arquivos ext2 não possui journaling e foi substituído pelo ext3.

## **Ext3**

O sistema de arquivos ext3 é uma versão do ext2 com suporte a journaling. Portanto, o ext3 tem as mesmas características do ext2, mas com suporte journaling.

Essa característica foi uma evolução e tornou o ext3 um sistema de arquivos muito estável e robusto.

Como no ext3 só foi adicionado o suporte a journaling, podemos converter um sistema de arquivos ext2 para ext3, adicionado suporte a journaling, e também podemos converter um sistema de arquivos ext3 para ext2, removendo o suporte a journaling.

## **ReiserFS**

O sistema de arquivos ReiserFS foi criado recentemente. Mas atualmente quase todas as distribuições Linux o suportam.

Sua performance é muito boa, principalmente para um número muito grande de arquivos pequenos.

ReiserFS também possui suporte a journaling.

## **XFS**

O sistema de arquivos XFS também possui suporte a journaling. Foi desenvolvido originalmente pela Silicon Graphics e posteriormente disponibilizado o código fonte. O XFS é considerado um dos melhores sistemas de arquivos para banco de dados, pois é muito rápido na gravação.

XFS utiliza muitos recursos de cache com memória RAM, e para utilizar XFS é recomendado utilizar sistemas que possuem redundância de energia.

## **SWAP**

SWAP é um espaço reservado para troca de dados com a memória RAM.

Em alguns lugares ele não é mencionado como um Sistema de Arquivos, mas resolvi descrever aqui, pois faz parte deste artigo.

## **VFAT**

O sistema de arquivos VFAT é também conhecido como FAT32 (M\$ Windows).

O sistema de arquivos VFAT não possui suporte a journaling. É utilizado normalmente para transferir dados entre sistemas M\$ Windows e o Linux instalados no mesmo disco, pois pode ser lido e escrito por ambos os sistemas operacionais.

O sistema de arquivos VFAT está longe de ser um sistema de arquivos utilizado para Sistemas Linux, exceto para compartilhamento/compatibilidade entre o M\$ Windows e Linux.

Se você utilizar VFAT no Linux, esteja certo de perder alguns atributos, tais como: permissão de execução, links simbólicos, entre outras coisas.

Ambos os sistemas de arquivos ext3 e ReiserFS são maduros o bastante para serem utilizados como padrão no Linux. Esses dois são os mais utilizados pelas distribuições Linux.

Os sistemas de arquivo Microsoft **FAT** e **NTFS** utilizados pelo Windows também são suportados pelo Linux.

### **Como é a estrutura de diretórios**

O sistema de arquivos (File System, em inglês) é uma estrutura composta de Arquivos e Diretórios. O sistema de arquivo, pelo menos teoricamente, é infinito. O limite é imposto somente pelo tamanho do disco rígido. O Linux utiliza um sistema hierárquico onde cada tipo de dados tem seu lugarzinho específico para ser guardado. Essa hierarquia é conhecida como “estrutura de árvore invertida”, e o topo é chamado de diretório raiz, que é representado por “/”.

Essa conversa poderia se estender a: “tipos de arquivos” e então teríamos “arquivos simples”, “arquivos-diretório”, “arquivos especiais”. É um assunto tão infinito quanto a teoria da própria estrutura.

### **Habitua-se!**

Quem cai de paraquedas no mundo do Linux fica em choque se perguntando se existe alguma coisa parecida com o Windows. No Windows, os arquivos do sistema ficam no diretório “Windows” e os programas em “Arquivos de Programas”. O Linux lida com isso ao contrário, no diretório raiz só existem as pastas do sistema e os seus arquivos pessoais são guardados dentro do diretório home, que é criada para cada usuário.

Dentro do Linux as partições, pen drivers, cdrom não aparecem como: C:\ , D:\ , E:\... Tudo faz parte do diretório raiz, ou seja, para o sistema todo mundo é diretório independente do dispositivo.

Uma descrição básica dos diretórios:

Vamos conhecê-los de acordo com a FHS (Filesystem Hierarchy Standard):

Todos os diretórios abaixo estão dentro do diretório raiz, ou seja, “/”.

/bin : Arquivos e programas do sistema que são usados com frequência pelos usuários.

/boot : Arquivos necessários para a inicialização do sistema.

/cdrom : Ponto de montagem da unidade de CD-ROM.

/dev : Arquivos usados para acessar dispositivos do computador.

/etc : Arquivos de configuração do computador.

/floppy : Ponto de montagem de unidade de disquetes

/home : Diretório que contém os arquivos de cada usuário.

/lib : Bibliotecas do sistema.

/lost+found : Local de arquivos e/ou diretórios recuperados pelo sistema.

/mnt : Ponto de montagem temporário.

/proc : Sistema de arquivos do Kernel.

/root : Diretório do usuário root, o administrador do sistema.

/opt : Local para aplicativos opcionais serem instalados.

/media : Ponto de montagem de mídia removível, câmeras digitais, pen drives

/sbin : Diretório de programas usados pelo super usuário (root) para administração e controle do funcionamento do sistema.

/tmp : Arquivos temporários criados por programas.

/usr : Diretório dos aplicativos. A maioria estará instalada neste diretório. Curiosidade: usr não quer dizer "User" e sim "Unix System Resources".

/var : Diretório contém arquivos que são gravados com frequência pelos aplicativos do sistema, como: e-mails, cache, spool de impressora.

Essa estrutura é considerada padrão. Encontrará a mesma se estiver utilizando a distribuição da Red Hat, SuSe ou o Ubuntu.

## **4. A 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 sistema operacional. O shell também implementa um linguagem simples de programação que permite o desenvolvimento de pequenos programas (os famosos shell scripts).

### **Um pouco de história**

Agora que nós já sabemos o que é o shell, vamos ver um pouco de sua história.

Desenvolvido por S.R Bourne em 1975, o Bourne Shell foi um dos primeiros shells



desenvolvidos. Por ser bastante simples ele continua sendo até hoje um dos mais rápidos e mais leves.

A evolução do Bourne Shell foi o C Shell. Desenvolvido por Bill Joy ele apresentava como inovações o histórico de comandos, o alias (que permite usar um "apelido" para comandos complexos) e o controle de processos em foreground e background. Mas a característica principal do C Shell era a semelhança de sua sintaxe com a da linguagem C. Recentemente tivemos o desenvolvimento do Turbo C Shell, que apresenta algumas melhorias em relação ao C Shell.

Na busca da implementação das características positivas do Bourne Shell e do C Shell, surgiu o Korn Shell, desenvolvido por David Korn.

O Bash (Bourne Again Shell) surgiu como uma reimplementação do Bourne Shell realizada pelo Projeto GNU ([www.gnuproject.org](http://www.gnuproject.org)). Apresentando melhorias em relação ao Bourne Shell, esse shell se popularizou pela expansão em sistemas **GNU/Linux**.

## 5. Trabalhando com arquivos e pastas.

*Escrito por Emerson Alecrim - Publicado em 05/12/2006 - Atualizado em 06/03/2010*

Quando um terminal é acessado, uma informação aparece no campo de inserção de comandos. É importante saber interpretá-la. Para isso, veja os exemplos abaixo:

Exemplo 1: **[root@infowester /root]#**

Exemplo 2: **[wester@alecrim /]\$**

Observação: dependendo de sua distribuição e de seu shell, a linha de comandos pode ter um formato ligeiramente diferente do que é mostrado nos exemplos. No Ubuntu Linux, por exemplo, o segundo exemplo fica na seguinte forma:

wester@alecrim: ~\$

Nos exemplos, a palavra existente antes do símbolo @ diz qual o nome do usuário que está usando o terminal. Os nomes que aparecem depois do @ indicam o computador que está sendo acessado seguido do diretório.

O caractere que aparece no final indica qual o "poder" do usuário. Se o símbolo for #, significa que usuário tem privilégios de administrador (root). Por outro lado, se o símbolo for \$, significa que este é um usuário comum, incapaz de acessar todos os recursos que um administrador acessa. Independente de qual seja, é depois do caractere que o usuário pode digitar os comandos.

## Os comandos básicos do Linux

A relação a seguir mostra os comandos seguidos de uma breve descrição:

**cal:** exibe um calendário;

**cat arquivo:** mostra o conteúdo de um arquivo. Por exemplo, para ver o arquivo infowester.txt, basta digitar cat infowester.txt;

**cd diretório:** abre um diretório. Por exemplo, para abrir a pasta /mnt, basta digitar cd /mnt. Para ir ao diretório raiz a partir de qualquer outro, digite apenas cd;

**chmod:** comando para alterar as permissões de arquivos e diretórios. Saiba mais neste artigo sobre permissões;

**clear:** elimina todo o conteúdo visível, deixando a linha de comando no topo, como se o terminal acabasse de ter sido acessado;

**cp origem destino:** copia um arquivo ou diretório para outro local. Por exemplo, para copiar o arquivo infowester.txt com o nome infowester2.txt para /home, basta digitar cp infowester.txt /home/infowester2.txt;

**date:** mostra a data e a hora atual;

**df:** mostra as partições usadas;

**diff arquivo1 arquivo2:** indica as diferenças entre dois arquivos, por exemplo: diff calc.c calc2.c;

**du diretório:** mostra o tamanho de um diretório;

**emacs:** abre o editor de textos emacs;

**file arquivo:** mostra informações de um arquivo;

**find diretório parâmetro termo:** o comando find serve para localizar informações. Para isso, deve-se digitar o comando seguido do diretório da pesquisa mais um parâmetro (ver lista abaixo) e o termo da busca. Parâmetros:

name - busca por nome

type - busca por tipo

size - busca pelo tamanho do arquivo

mtime - busca por data de modificação

Exemplo: find /home name tristania

**finger usuário:** exibe informações sobre o usuário indicado;

**free:** mostra a quantidade de memória RAM disponível;

**halt:** desliga o computador;

**history:** mostra os últimos comandos inseridos;

**kill:** encerra processos em andamento. Saiba mais no artigo Processos no Linux;

**ls:** lista os arquivos e diretórios da pasta atual;

**lpr arquivo:** imprime o arquivo especificado;

**lpq:** mostra o status da fila de impressão;

**lprm:** remove trabalhos da fila de impressão;

**lynx:** abre o navegador de internet de mesmo nome;

**mv origem destino:** tem a mesma função do comando cp, só que ao invés de copiar, move o arquivo ou o diretório para o destino especificado;

**mkdir diretório:** cria um diretório, por exemplo, mkdir infowester cria uma pasta de nome infowester;

**pwd:** mostra o diretório em que você está;

**reboot:** reinicia o sistema imediatamente (pouco recomendável, preferível shutdown -r now);

**rm arquivo:** apaga o arquivo especificado;

**rmdir diretório:** apaga o diretório especificado, desde que vazio;

**shutdown:** desliga ou reinicia o computador, veja:

shutdown -r now: reinicia o computador

shutdown -h now: desliga o computador

O parâmetro **now** pode ser mudado. Por exemplo: digite shutdown -r +10 e o sistema irá reiniciar daqui a 10 minutos;

**su:** passa para o usuário administrador, isto é, root (perceba que o símbolo \$ mudará para #);

**tar -xvzf arquivo.tar.gz:** extrai um arquivo compactado em tar.gz.

**telnet:** ativa o serviço de Telnet em uma máquina. Para acessar esse computador a partir de outros por Telnet, basta digitar telnet nomedamáquina ou telnet IP. Por exemplo: telnet 192.168.0.10. Após abrir o Telnet, digite help para conhecer suas funções;

**top:** exibe a lista dos processos, conforme os recursos de memória consumidos;

**uname:** mostra informações do sistema operacional e do computador. Digite uname -a para obter mais detalhes;

**vi:** inicia o editor de textos vi.

**uptime:** mostra a quantas horas seu computador está ligado;

**whereis nome:** procura pelo binário do arquivo indicado, útil para conhecer seu diretório ou se ele existe no sistema;

### **Considerações:**

Praticamente todos os comandos citados possuem parâmetros que permitem incrementar suas funcionalidades. Por exemplo, se você digitar o comando ls com o parâmetro -R (ls -R), este mostrará todos os arquivos do diretório, inclusive os ocultos.

A melhor forma de conhecer os parâmetros adicionais de cada comando é consultando

as informações de ajuda. Para isso, pode-se usar o recurso `--help`. Veja o exemplo para o comando `ls`:

### **ls --help**

.: Livros sugeridos :.  
:: Linux - O guia essencial  
:: Comandos do Linux - Prático e Didático  
:: Manual completo do Linux - Guia do administrador  
Via Shopping UOL

Também é possível utilizar o comando `man` (desde que seu conteúdo esteja instalado), que geralmente fornece informações mais detalhadas. Para usar o `man` para obter detalhes do comando `cp`, por exemplo, a sintaxe é:

### **man cp**

Se você estiver utilizando o `bash`, pode-se aplicar o comando `help` ou `info` da mesma forma que o comando `man`:

### **help cp**

### **info cp**

Assim como conhecer os comandos básicos do Linux é importante, também o é saber como acessar seus recursos de ajuda, pois isso te desobriga de decorar as sequências das funcionalidades extras. Sabendo usar todos os recursos, você certamente terá boa produtividade em suas tarefas no Linux.

## **7. Administração de contas de usuários e grupos.**

Abaixo alguns comandos para a administração de contas de usuários e grupos.

**id usuário:** mostra qual o número de identificação do usuário especificado no sistema;

**passwd:** altera sua senha. Para um administrador mudar a senha de um usuário, basta digitar `passwd` seguido do nome deste;

**ps:** mostra os processos em execução.

**useradd usuário:** cria uma nova conta usuário, por exemplo, `useradd marvin` cria o usuário marvin;

**userdel usuário:** apaga a conta do usuário especificado;

**w:** mostra os usuários logados atualmente no computador (útil para servidores);

**who:** mostra quem está usando o sistema.

## **8. O ambiente gráfico e seus aplicativos.**

O ambiente gráfico do Linux é chamado de **X Windows System** ou como é mais conhecido como **X11** ou apenas **X**. Ele é a interface completa para Linux e Unix. O X Window System, foi criado no Instituto de Tecnologia de Massachusetts (MIT). O X é um sistema de janelas.

O X tem todas as suas especificações abertas por isso tem grande sucesso entre os fabricantes de hardware e software que criaram um conjunto de implementações abertas.

Veja o exemplo de vários ambientes gráficos:

### KDE no Mandriva 2006.



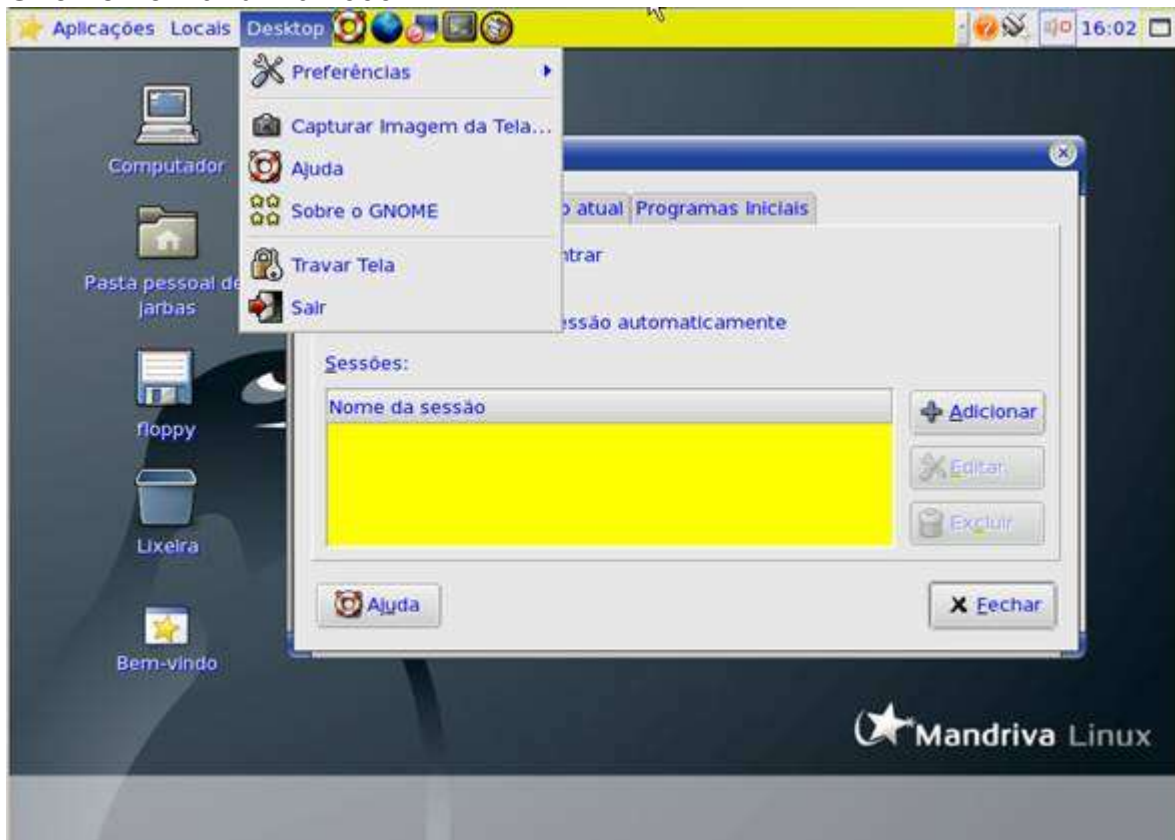
### Kde no SUSE



## Kde no Conectiva 10



## Gnome no Mandriva 2006





## Vantagem do X Windows System

A grande vantagem do X Windows é que ele poder Cliente/Servidor ou seja. Ele pode ser executado tanto localmente como remotamente. Digamos que você tenha uma aplicação em algum Linux que todas as outras estações precisam executá-las também. Você pode configurar para que estas estações acessem o Servidor X onde está esta aplicação. As estações podem funcionar apenas localmente, remotamente ou das duas formas. Quem vai determinar isso é o administrador.

Uma estação antiga com poucos recursos, pode ser configurada para acessar o ambiente gráfico num Servidor X Windows System, pois o processamento é feito no Servidor e não na estação.

**Obs1:** O recurso de acessar o Linux remotamente se for comparar com o mundo Windows Server seria como o Terminal Service.

Uma outra vantagem é que se uma aplicação foi feita no Gnome e outra no KDE, é possível abrir os dois na mesma estação ou servidor remoto.

## Apresentando o XFree86

O XFree86 é um porte de livre distribuição do sistema X window e ele implementa tanto o servidor quanto o cliente X.

É através do XFree86 que é dado o suporte de:

- » Bibliotecas
- » Utilitários de configuração e os "drivers"

O XFree86 faz o desenho na tela mas o ambiente gráfico, mas o responsável pelo gerenciamento de janelas é feito por programas como o KDE, GNOME.

O gerenciador de login usado é o **KDM** (KDE Display Manager).

## Acessando vários terminais gráficos

Em qualquer Linux por default, vem configurado para você acessar através do Modo Texto até seis terminais.

Terminal 1	Alt+F1
Terminal 2	Alt+F2
Terminal 3	Alt+F3
Terminal 4	Alt+F4
Terminal 5	Alt+F5
Terminal 6	Alt+F6

De um terminal no Modo Texto para voltar para o Modo Gráfico basta pressionar **Alt+F7**.

Voltando do Modo Gráfico para o Modo Texto, é só pressionar Ctrl+Alt+F1 até Ctrl+Alt+F6.

Segundo esta lógica, é possível configurar os outros terminais do F8 até F12. E é o que vamos aprender agora.

Vamos ao primeiro exemplo:

1. Estando no Modo Texto em qualquer Linux que tem o X configurado digite para o Kde.

**startx -- :2**

Será aberta a tela do KDE com o usuário que estava logado no terminal do Modo Texto.

2. Abrindo outro terminal:

**startx -- :3.**

**Obs: Abrir vários servidores X, consome recurso da máquina.**

Até agora, fizemos apenas acessos locais.

### **Configurando o Servidor X para ser acessado remotamente**

Para ativar um Servidor X para acesso remoto, precisamos que você altere dois arquivos apenas (Xaccess e o kdmrc). O kdmrc. No Mandriva 2006, uns dos arquivos já vem prontos. Mas no Conectiva não. Então, vou mostrar os passos.

1. No Modo Texto digite `vi /etc/X11/xdm/Xaccess`. No Mandriva 2006 esta linha já está descomentada, mas no Conectiva 10 não.

### **Configurando o Servidor X para ser acessado remotamente**

Tutorial para ser feito como teste – Não é obrigatório somente um exercício a mais.

Por: Jarbas Teixeira. (<http://www.juliobattisti.com.br/tutoriais/jarbasteixeira/linux001.asp>)

Para ativar um Servidor X para acesso remoto, precisamos que você altere dois arquivos apenas (Xaccess e o kdmrc). O kdmrc. No Mandriva 2006, uns dos arquivos já vem prontos. Mas no Conectiva não. Então, vou mostrar os passos.

1. No Modo Texto digite `vi /etc/X11/xdm/Xaccess`. No Mandriva 2006 esta linha já está descomentada, mas no Conectiva 10 não.

```
*      #any host can get a login window
```

2. A outra linha é:

```
* CHOOSER BROADCAST #any indirect host can get a chooser
```

No Mandriva 2006 esta linha já está descomentada, mas no Conectiva 10 não.

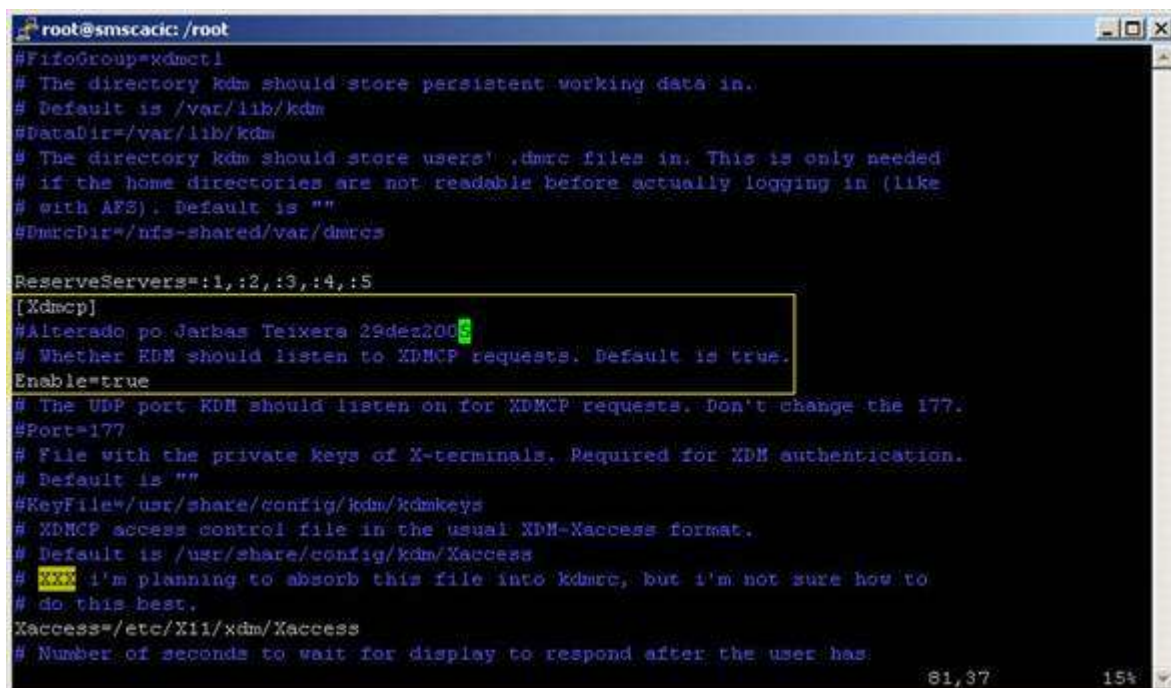
Vamos agora ao outro arquivo. É o arquivo **kdmrc**.

3. `vi /etc/kde/kdm/kdmrc`.

### **Na seção Xdmcp**

Mude de **Enable=false** para **Enable=true**

Veja a ilustração:



```
root@smcadi: /root
#FifoGroup=xmcp
# The directory kdm should store persistent working data in.
# Default is /var/lib/kdm
#DataDir=/var/lib/kdm
# The directory kdm should store users' .dmrc files in. This is only needed
# if the home directories are not readable before actually logging in (like
# with AFS). Default is ""
#DmrcDir=/nfs-shared/var/dmrcs

ReserveServers=:1,:2,:3,:4,:5
[Xdmcp]
#Alterado por Jarbas Teixeira 29dez2008
# Whether KDM should listen to XDMCP requests. Default is true.
Enable=true
# The UDP port KDM should listen on for XDMCP requests. Don't change the 177.
#Port=177
# File with the private keys of X-terminals. Required for XDM authentication.
# Default is ""
#KeyFile=/usr/share/config/kdm/kdmkeys
# XDMCP access control file in the usual XDM-Xaccess format.
# Default is /usr/share/config/kdm/Xaccess
# XXXX I'm planning to absorb this file into kdmrc, but I'm not sure how to
# do this best.
Xaccess=/etc/X11/xdm/Xaccess
# Number of seconds to wait for display to respond after the user has
```

**Dica:** Basicamente nas distribuições que usam o KDE os arquivos são os mesmos só estão em diretórios diferentes. No Suse e Conectiva por exemplo, os arquivos estão em locais diferentes mas os procedimentos são os mesmos e funcionou seguindo os mesmos passos.

4. Agora reboot o servidor e vamos começar a acessá-lo.

O reboot não é necessário. Existem alguns comandos que fazem isto.

### Acessando o servidor através do Linux

1. Noutra estação Linux, digite no Modo texto:

```
X :2 -query ip_do_servidor
```

Aguarde um pouco enquanto o servidor é acessado.

2. Para acessá-lo digite **Ctrl Alt F8**.

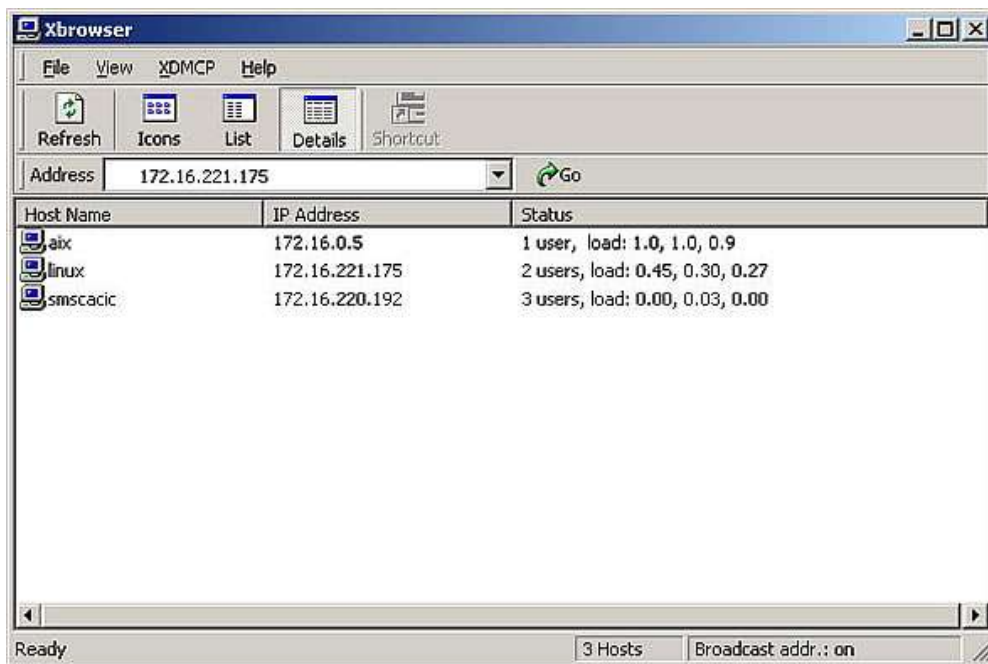
É só isso. Simples e prático.

### Acessando o servidor através do Windows

No Windows você tem que instalar um programa para fazer isso. Aqui na empresa usamos o Xmanager tanto para acessar os Linux quanto os UNIX.

1. Depois de instalado, execute o programa **Xbrowser**.

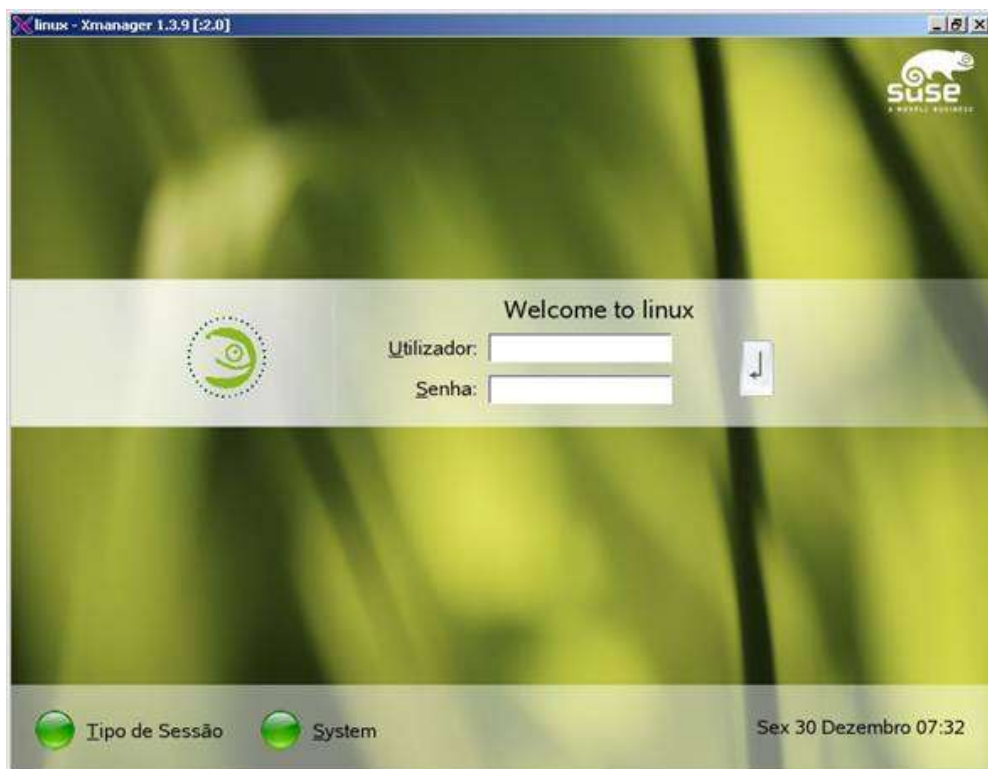
Veja a ilustração:



Ele mostra um IBM AIX e dois Linux.

Para acessá-lo, é só clicar duas vezes em qualquer conexão. Veja a ilustração:

Tela do Suse de dentro do Windows:



Veja a tela do Mandriva:



## Fazendo um broadcast na red

Existe um comando que lista todos os servidores X da rede.

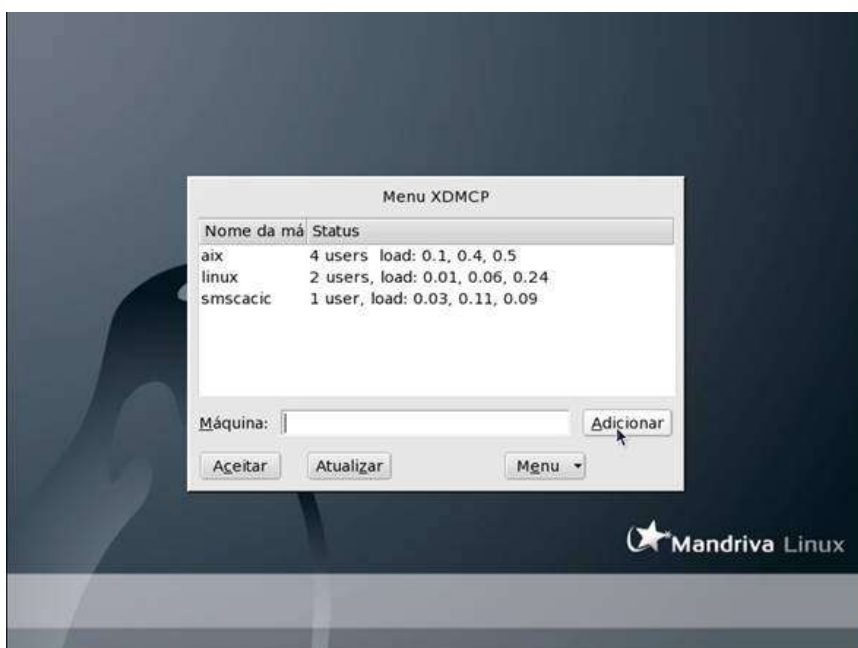
### 1. Digite no Modo Texto **X :2 -indirect 172.16.0.0**

Este comando faz com que seja listado todos os servidores X da rede. O detalhe é que o cliente não fica dependendo apenas de um servidor. Sempre será exibido a lista.

E, será possível escolher em qual se deseja conectar.

A próxima ilustração, eu estava no Conectiva 10 e acessei a lista de servidores X que estão configurados aqui na minha rede. Tinha um Suse, um Mandriva e um IBM AIX.

Veja:



## 9. Administração de redes em software livre.

Hoje em dia, quase todo mundo que tem acesso via banda larga e tem mais do que um PC em casa, acaba de uma forma ou de outra configurando um deles para compartilhar recursos na rede, ou compartilhar a conexão. Conforme os upgrades e trocas acontecem, mais cedo ou mais tarde você acaba ficando com um micro disponível e acaba usando-o para configurar um servidor de arquivos, ou outro tipo de servidor dedicado para a rede, sem falar no casos em que você monta uma máquina especialmente para a tarefa.

Neste curso só falaremos das possibilidades de gerenciar uma rede utilizando Linux, para mais informações acesse o tutorial completo no guia do hardware (<http://www.hardware.com.br/tutoriais/servidor-rede-local-ubuntu/>) Configurando um servidor de rede com o Ubuntu.

Um servidor de rede Linux é capaz de:

- Compartilhar arquivos e impressoras através do Samba, servindo como um servidor de arquivos para a rede local.
- Rodar máquinas virtuais através do VMware Server, que ficarão acessíveis para toda a rede (você pode manter uma VM com o Windows caso precise rodar aplicativos para a plataforma, por exemplo e acessá-la de qualquer um dos micros da rede).
- Servidor DHCP.
- Compartilhamento da conexão e proxy transparente com o Squid.
- Servidor SSH e NX Server, para que você possa acessar o servidor remotamente.
- Domínio virtual no No-IP, para que o servidor tenha um endereço fixo, para acesso remoto.

E muito mais.

## 10. Aplicativos de apoio (S.O. de apoio: Linux)

O próximo tutorial foi retirado do endereço: (<http://gaigalas.net/Artigos/UbuntuServidorPHPMySQL.html>)

### **Tutorial de Instalação do Apache, PHP e MySQL no Ubuntu**

por [Alexandre Gomes Gaigalas](#), do [Acorda pra Web!](#)

#### **Nesse Tutorial você vai aprender...**

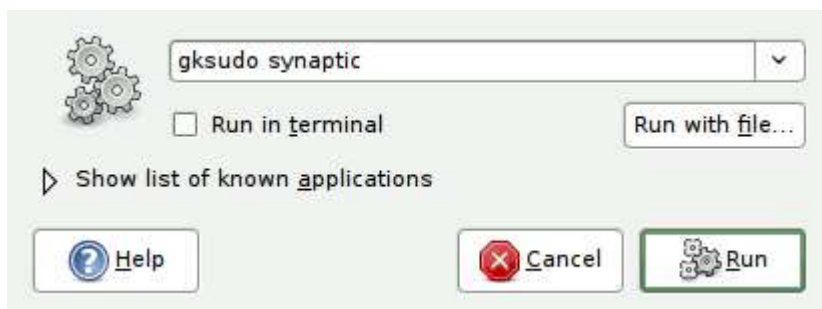
- Básico da instalação de pacotes no Ubuntu
- Instalação e estrutura básica do Apache 2 no Ubuntu 8.04
- Instalação do PHP5
- Instalação do MySQL

#### **Instalação de pacotes no Ubuntu**

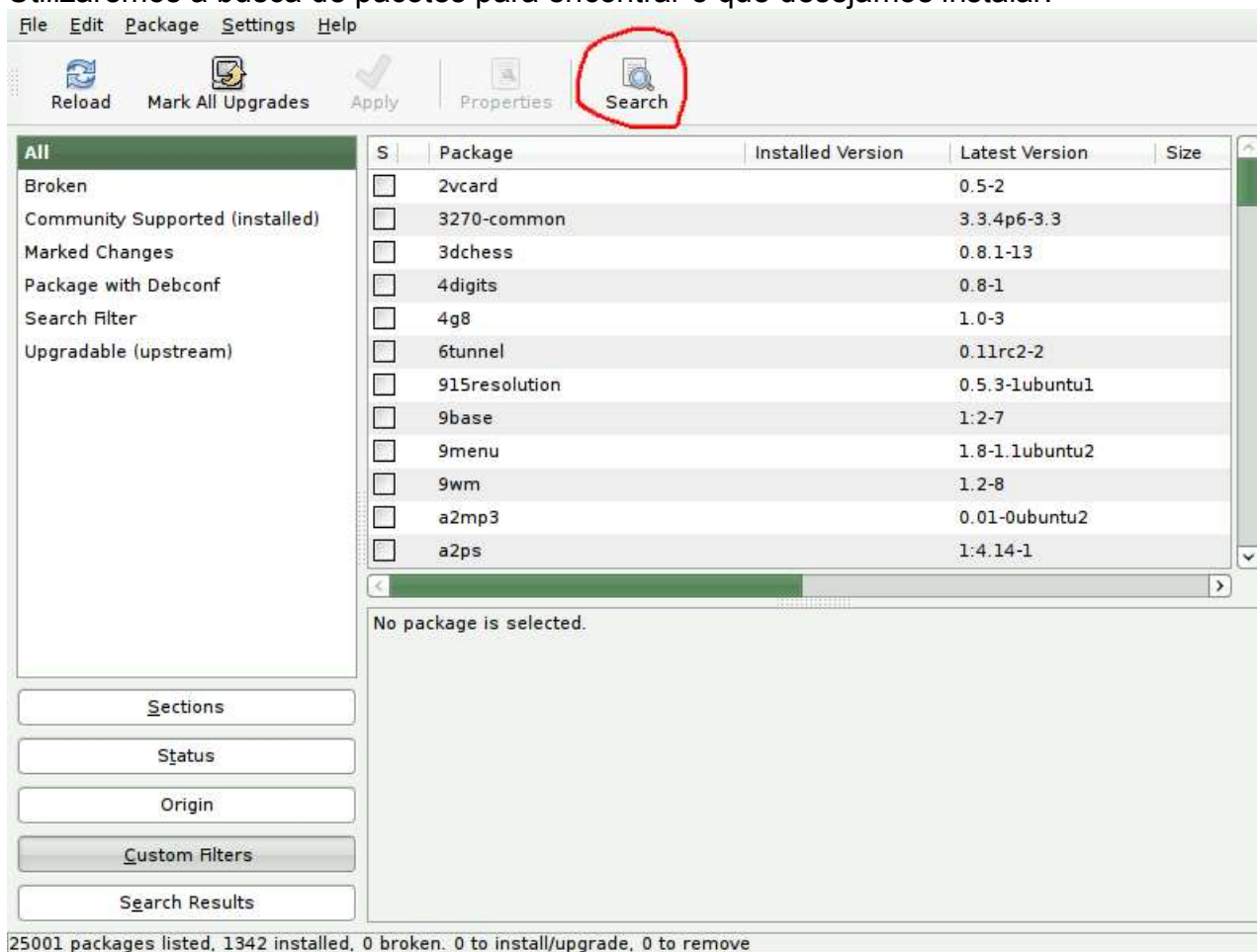
Distribuições Linux, em sua maioria, gerenciam os aplicativos utilizando pacotes que podem facilmente ser baixados, instalados e configurados no ambiente do sistema operacional.

#### **Gerenciador de Pacotes Synaptic**

O Ubuntu, por padrão, vem com o excelente gerenciador de pacotes Synaptic. Para iniciar o Synaptic, basta acessar o menu *Sistema > Administração > Gerenciador de Pacotes Synaptic*, ou apertar Alt+F2 e digitar o comando `gksudo synaptic`.



A Interface do Synaptic é bem simples. No centro existe uma lista de pacotes que corresponde ao filtro no painel da esquerda. Para cada pacote, há a opção de *marcar para instalação* e *marcar para remoção*. Na barra de ferramentas há um botão para recarregar a lista de pacotes, um para marcar os pacotes que possuem *upgrade*, um para aplicar as alterações marcadas, um para ver propriedades dos pacotes e um para busca. Utilizaremos a busca de pacotes para encontrar o que desejamos instalar:



### Pacotes do Apache

Faremos uma busca por **apache2** e marcaremos os seguintes pacotes: **apache2** e **libapache2-mod-php5**.



FileEditPackageSettingsHelp

Reload

Mark All Upgrades

Apply

Properties

Search

All

apache2

S	Package	Installed Version	Latest Version	Size	Desc
<input type="checkbox"/>	adzapper		20070317-2		prox
<input checked="" type="checkbox"/>	apache2		2.2.8-1ubuntu0.2		Next
<input checked="" type="checkbox"/>	apache2.2-common		2.2.8-1ubuntu0.2		Next
<input type="checkbox"/>	apache2-doc		2.2.8-1ubuntu0.2		docu
<input type="checkbox"/>	apache2-mpm-event		2.2.8-1ubuntu0.2		Even
<input type="checkbox"/>	apache2-mpm-itk		2.2.6-01-1build2.1		mult
<input type="checkbox"/>	apache2-mpm-perchild		2.2.8-1ubuntu0.2		Tran
<input type="checkbox"/>	apache2-mpm-prefork		2.2.8-1ubuntu0.2		Tradi
<input checked="" type="checkbox"/>	apache2-mpm-worker		2.2.8-1ubuntu0.2		High
<input type="checkbox"/>	apache2-prefork-dev		2.2.8-1ubuntu0.2		deve
<input type="checkbox"/>	apache2-src		2.2.8-1ubuntu0.2		Apac
<input type="checkbox"/>	apache2-threaded-dev		2.2.8-1ubuntu0.2		deve
<input checked="" type="checkbox"/>	apache2-utils		2.2.8-1ubuntu0.2		utility
<input type="checkbox"/>	certbot		0.22.2		stre

Sections

Status

Origin

Custom Filters

Search Results

Next generation, scalable, extendable web server

Apache v2 is the next generation of the omnipresent Apache web server. This version - a total rewrite - introduces many new improvements, such as threading, a new API, IPv6 support, request/response filtering, and more.

94 packages listed, 1342 installed, 0 broken. 4 to install/upgrade, 0 to remove; 4436 kB will be used

FileEditPackageSettingsHelp

Reload

Mark All Upgrades

Apply

Properties

Search

All

apache2

S	Package	Installed Version	Latest Version	Size	Desc
<input type="checkbox"/>	libapache2-mod-perl2-doc		2.0.3-2ubuntu2		Integ
<input checked="" type="checkbox"/>	libapache2-mod-php5		5.2.4-2ubuntu5.1		serv
<input type="checkbox"/>	libapache2-mod-proxy-html		3.0.0-1		Apac
<input type="checkbox"/>	libapache2-mod-python		3.3.1-2build1		Apac
<input type="checkbox"/>	libapache2-mod-python-doc		3.3.1-2build1		Apac
<input type="checkbox"/>	libapache2-mod-random		2.1-0ubuntu1		Crea
<input type="checkbox"/>	libapache2-mod-removeip		1.0b-5		Modu
<input type="checkbox"/>	libapache2-mod-rpaf		0.5-3		mod
<input type="checkbox"/>	libapache2-mod-ruby		1.2.6-1.2		Emb
<input type="checkbox"/>	libapache2-mod-scgi		1.12-0.2		Apac
<input type="checkbox"/>	libapache2-mod-shib		1.3f.dfsg1-6		Shib
<input type="checkbox"/>	libapache2-mod-speedycgi		2.22-8		apac
<input type="checkbox"/>	libapache2-mod-suphp		0.6.2-2ubuntu1		Apac
<input type="checkbox"/>	libapache2-mod-vhost-hash-alias		1.0-2		Fast

Sections

Status

Origin

Custom Filters

Search Results

server-side, HTML-embedded scripting language (apache 2 module)

This package provides the PHP5 module for the Apache 2 webserver (as found in the apache2-mpm-prefork package). Please note that this package ONLY works with Apache's prefork MPM, as it is not compiled thread-safe.

The following extensions are built in: bcmath bz2 calendar ctype date dba dom exif filter ftp gettext hash iconv json libxml mbstring mime\_magic openssl pcre posix Reflection session shmop SimpleXML soap sockets SPL

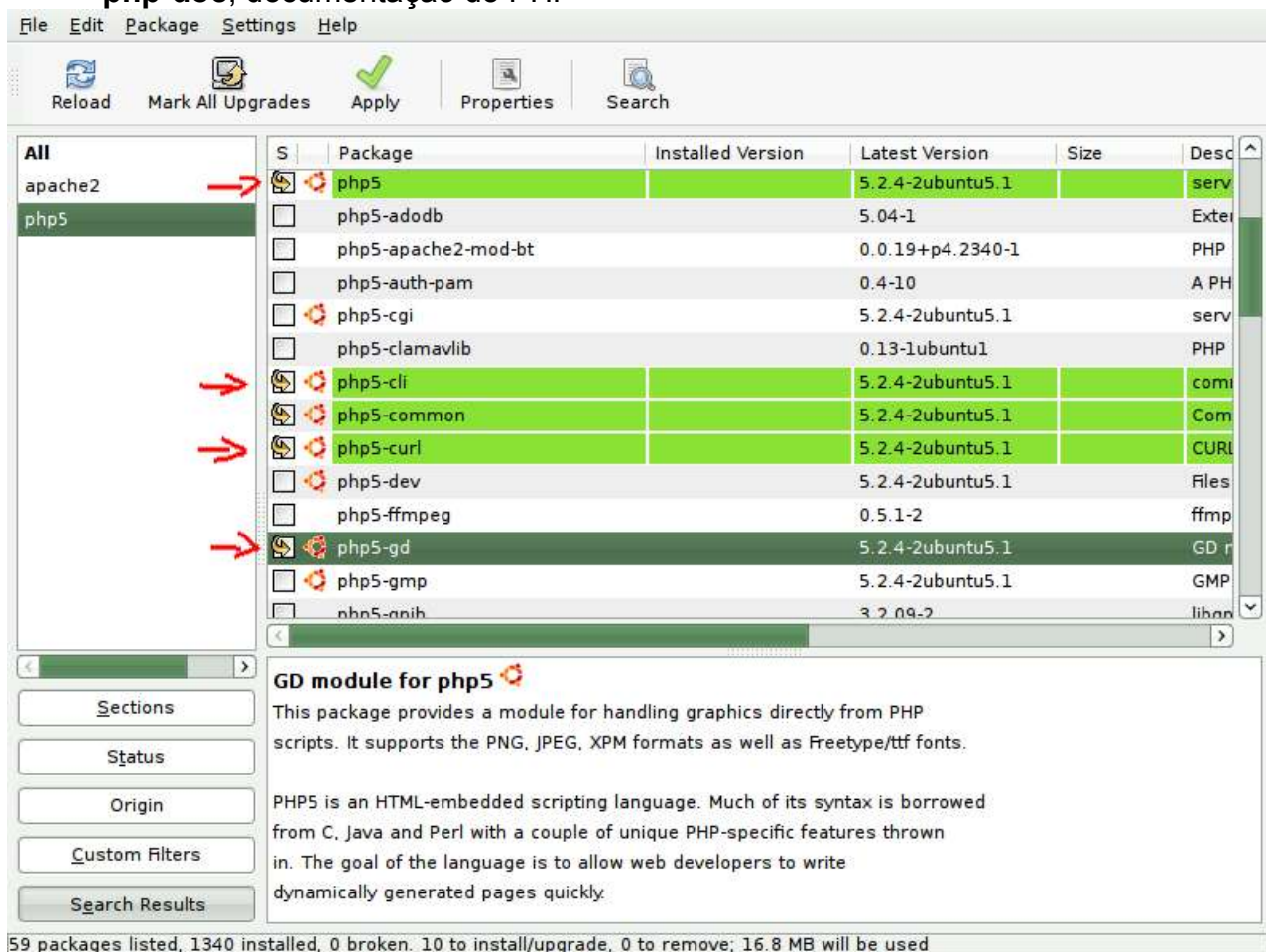
94 packages listed, 1342 installed, 0 broken. 6 to install/upgrade, 0 to remove; 10.9 MB will be used

Note que além desses pacotes que escolhemos, o Synaptic marca alguns outros. Essas são as dependências dos pacotes, ou seja, pacotes-filho necessários para que os pacotes-pai escolhidos funcionem.

### Pacotes do PHP

Depois, buscaremos por **php5** e marcaremos os pacotes:

- **php5**, o pacote principal
- **php5-cli**, para PHP em linha de comando
- **php5-common**, uma biblioteca raiz do PHP
- **php5-curl**, para biblioteca CURL
- **php5-gd**, biblioteca manipulação de imagens via PHP
- **php5-mysql**, para conexão com bancos de dados MySQL
- **php5-pgsql**, para conexão com bancos de dados PostgreSQL
- **php5-sqlite**, para conexão com bancos de dados SQLite versão 2
- **php5-sqlite3**, para conexão com bancos de dados SQLite versão 3
- **php5-sybase**, para conexão com Sybase e Microsoft SQL Server
- **php5-xsl**, para aplicação de estilos XSL
- **php-doc**, documentação do PHP



The screenshot shows the Synaptic Package Manager window. The left sidebar has 'All' and 'php5' listed. The main table displays the following packages:

S	Package	Installed Version	Latest Version	Size	Desc
<input checked="" type="checkbox"/>	php5		5.2.4-2ubuntu5.1		serv
<input type="checkbox"/>	php5-adodb		5.04-1		Exter
<input type="checkbox"/>	php5-apache2-mod-bt		0.0.19+p4.2340-1		PHP
<input type="checkbox"/>	php5-auth-pam		0.4-10		A PH
<input checked="" type="checkbox"/>	php5-cgi		5.2.4-2ubuntu5.1		serv
<input type="checkbox"/>	php5-clamavlib		0.13-1ubuntu1		PHP
<input checked="" type="checkbox"/>	php5-cli		5.2.4-2ubuntu5.1		comi
<input checked="" type="checkbox"/>	php5-common		5.2.4-2ubuntu5.1		Com
<input checked="" type="checkbox"/>	php5-curl		5.2.4-2ubuntu5.1		CURL
<input type="checkbox"/>	php5-dev		5.2.4-2ubuntu5.1		Files
<input type="checkbox"/>	php5-ffmpeg		0.5.1-2		ffmp
<input checked="" type="checkbox"/>	php5-gd		5.2.4-2ubuntu5.1		GD r
<input type="checkbox"/>	php5-gmp		5.2.4-2ubuntu5.1		GMP
<input type="checkbox"/>	php5-imap		3.2.09-2		liban

The bottom panel shows details for 'GD module for php5':

**GD module for php5**

This package provides a module for handling graphics directly from PHP scripts. It supports the PNG, JPEG, XPM formats as well as Freetype/ttf fonts.

PHP5 is an HTML-embedded scripting language. Much of its syntax is borrowed from C, Java and Perl with a couple of unique PHP-specific features thrown in. The goal of the language is to allow web developers to write dynamically generated pages quickly.

59 packages listed, 1340 installed, 0 broken. 10 to install/upgrade, 0 to remove; 16.8 MB will be used

File Edit Package Settings Help

Reload Mark All Upgrades Apply Properties Search

**All**  
apache2  
php5

S	Package	Installed Version	Latest Version	Size	Desc
<input type="checkbox"/>	php5-mhash		5.2.4-2ubuntu5.1		MHA:
<input type="checkbox"/>	php5-ming		1:0.3.0-12ubuntu1		Ming
<input checked="" type="checkbox"/>	php5-mysql		5.2.4-2ubuntu5.1		MySt
<input type="checkbox"/>	php5-odbc		5.2.4-2ubuntu5.1		ODB
<input checked="" type="checkbox"/>	php5-pgsql		5.2.4-2ubuntu5.1		Post
<input type="checkbox"/>	php5-pspell		5.2.4-2ubuntu5.1		pspe
<input type="checkbox"/>	php5-recode		5.2.4-2ubuntu5.1		reco
<input type="checkbox"/>	php5-sasl		0.1.0-1		Cyru
<input type="checkbox"/>	php5-snmpp		5.2.4-2ubuntu5.1		SNM
<input checked="" type="checkbox"/>	php5-sqlite		5.2.4-2ubuntu5.1		SQLI
<input checked="" type="checkbox"/>	php5-sqlite3		0.5-1		SQLI
<input type="checkbox"/>	php5-sqlrelay		1:0.38-3ubuntu1		SQL
<input type="checkbox"/>	php5-suhosin		0.9.22-1		adva

**PostgreSQL module for php5**

This package provides a module for PostgreSQL database connections directly from PHP scripts. It also includes the pdo\_pgsql module for use with the PHP Data Object extension.

PHP5 is an HTML-embedded scripting language. Much of its syntax is borrowed from C, Java and Perl with a couple of unique PHP-specific features thrown in. The goal of the language is to allow web developers to write dynamically generated pages quickly.

59 packages listed, 1340 installed, 0 broken. 17 to install/upgrade, 0 to remove; 20.9 MB will be used

File Edit Package Settings Help

Reload Mark All Upgrades Apply Properties Search

**All**  
apache2  
php5

S	Package	Installed Version	Latest Version	Size	Desc
<input type="checkbox"/>	php5-suhosin		0.9.22-1		adva
<input checked="" type="checkbox"/>	php5-sybase		5.2.4-2ubuntu5.1		Syba
<input type="checkbox"/>	php5-syck		0.55+svn270-1		YAML
<input type="checkbox"/>	php5-tidy		5.2.4-2ubuntu5.1		tidy
<input type="checkbox"/>	php5-uuid		1.5.1-1.1ubuntu1		OSS
<input type="checkbox"/>	php5-xapian		1.0.5-1		Xapi
<input type="checkbox"/>	php5-xcache		1.2.1-3		Fast
<input type="checkbox"/>	php5-xdebug		2.0.2-0ubuntu5		xdeb
<input type="checkbox"/>	php5-xmlrpc		5.2.4-2ubuntu5.1		XML-
<input checked="" type="checkbox"/>	php5-xsl		5.2.4-2ubuntu5.1		XSL
<input checked="" type="checkbox"/>	php-doc		20070501-1		Docu
<input type="checkbox"/>	php-image-canvas		0.2.2-1		Imag
<input type="checkbox"/>	php-image-graph		0.7.1-1		Imag
<input type="checkbox"/>	php-imlib		0.7.1		num

**Documentation for PHP4 and PHP5**

This package provides the documentation for the PHP scripting language.

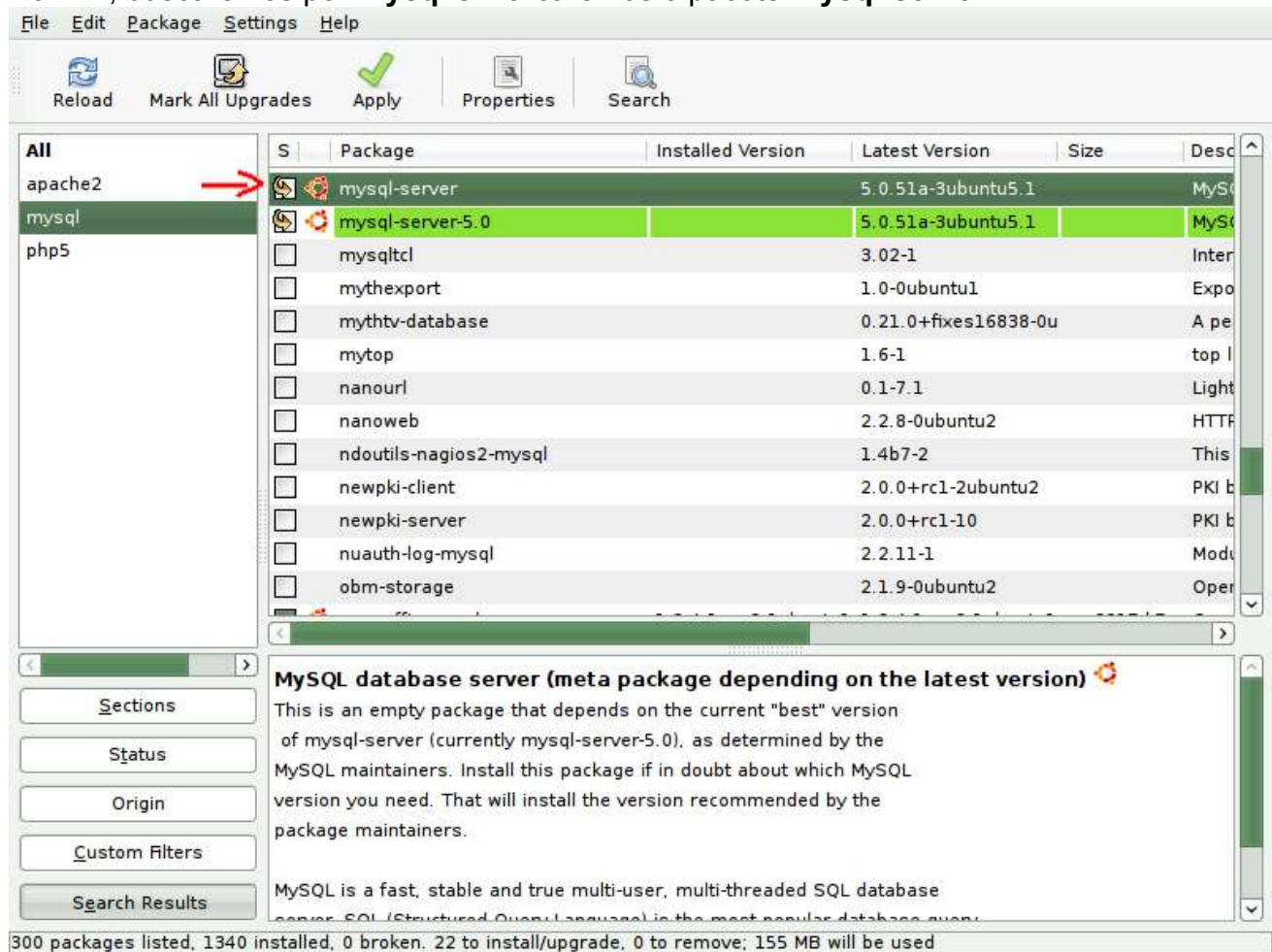
PHP, which stands for "PHP: Hypertext Preprocessor" is a widely-used Open Source general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. Its syntax draws upon C, Java, and Perl, and is easy to learn. The main goal of the language is to allow web developers to write dynamically generated webpages quickly, but you can do much more with PHP.

59 packages listed, 1340 installed, 0 broken. 20 to install/upgrade, 0 to remove; 68.3 MB will be used

**Pacotes do MySQL**

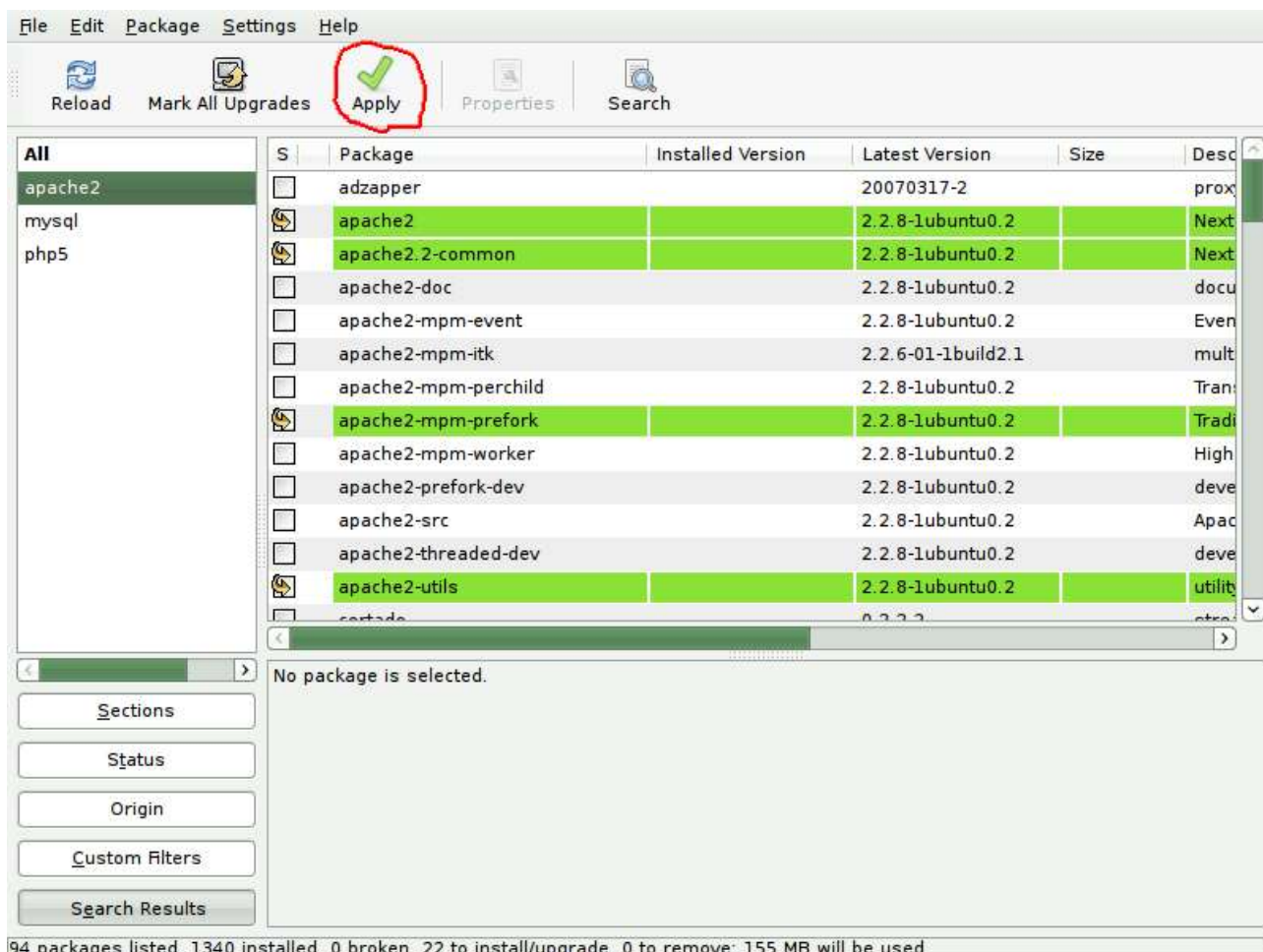


Por fim, buscaremos por **mysql** e marcaremos o pacote **mysql-server**.



## Finalizando a Instalação

Para completar a instalação, aplicaremos as mudanças no botão **"Apply"** (Aplicar):

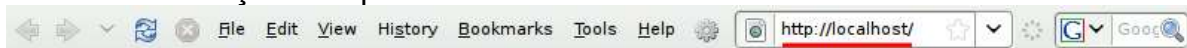


Após confirmar a instalação dos pacotes, o Synaptic fará uma conexão com o servidor do Ubuntu, baixará, instalará e configurará os pacotes selecionados. Durante a instalação, o configurador do MySQL pedirá uma senha, necessária para o usuário **root** do banco:

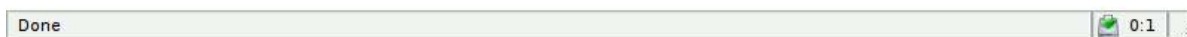


Digite a senha que preferir e prossiga, confirmando a senha e terminando a instalação.

Se tudo ocorreu bem, ao acessar o endereço <http://localhost/> você deverá ver a simpática tela de confirmação do Apache:



## It works!

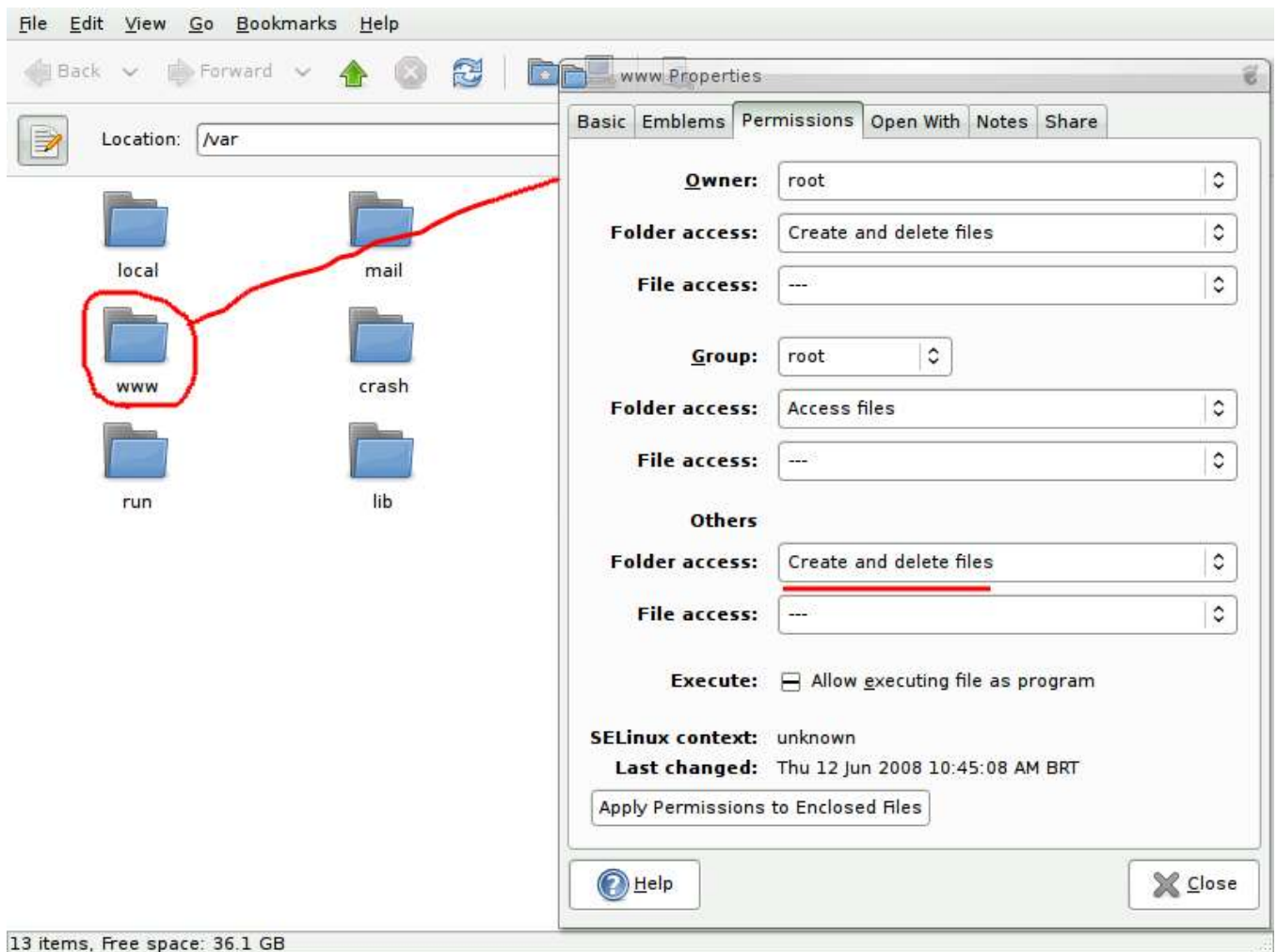


### Configuração Básica

Os arquivos de publicação desse servidor ficam na pasta **/var/www**. Precisaremos adicionar permissões para que você possa alterar o conteúdo dessa pasta. Para isso, abriremos o nautilus (gerenciador de arquivos do Ubuntu) como administrador:



Digite Alt+F2. O comando gksudo é o comando padrão para executar tarefas como administrador. Após a tela do nautilus abrir, abriremos a tela de propriedades da pasta **www** e colocaremos as permissões para que qualquer pessoa possa criar e excluir arquivos:



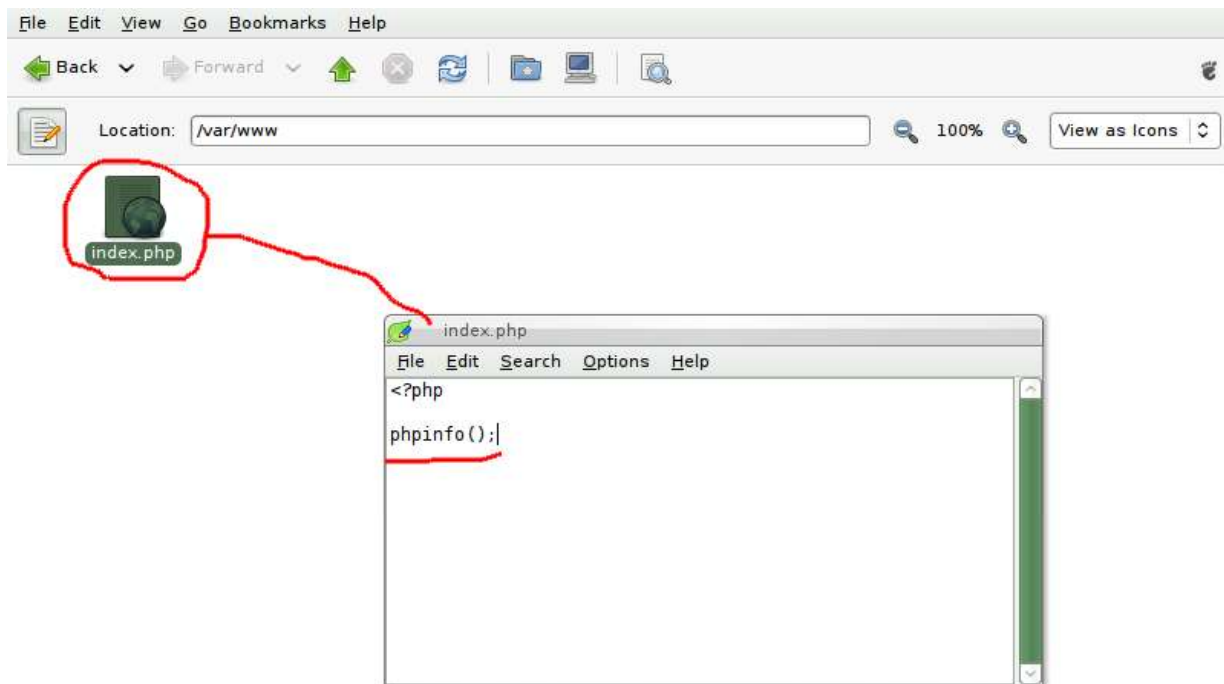
## Testando o PHP

Para testarmos o PHP, substituiremos o arquivo index.html por um arquivo **index.php** contendo o seguinte código:

```
<?php
```

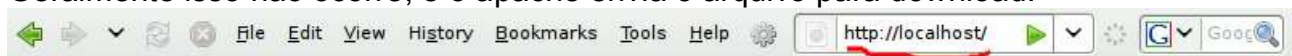
```
phpinfo();
```





"index.php" selected (17 bytes)

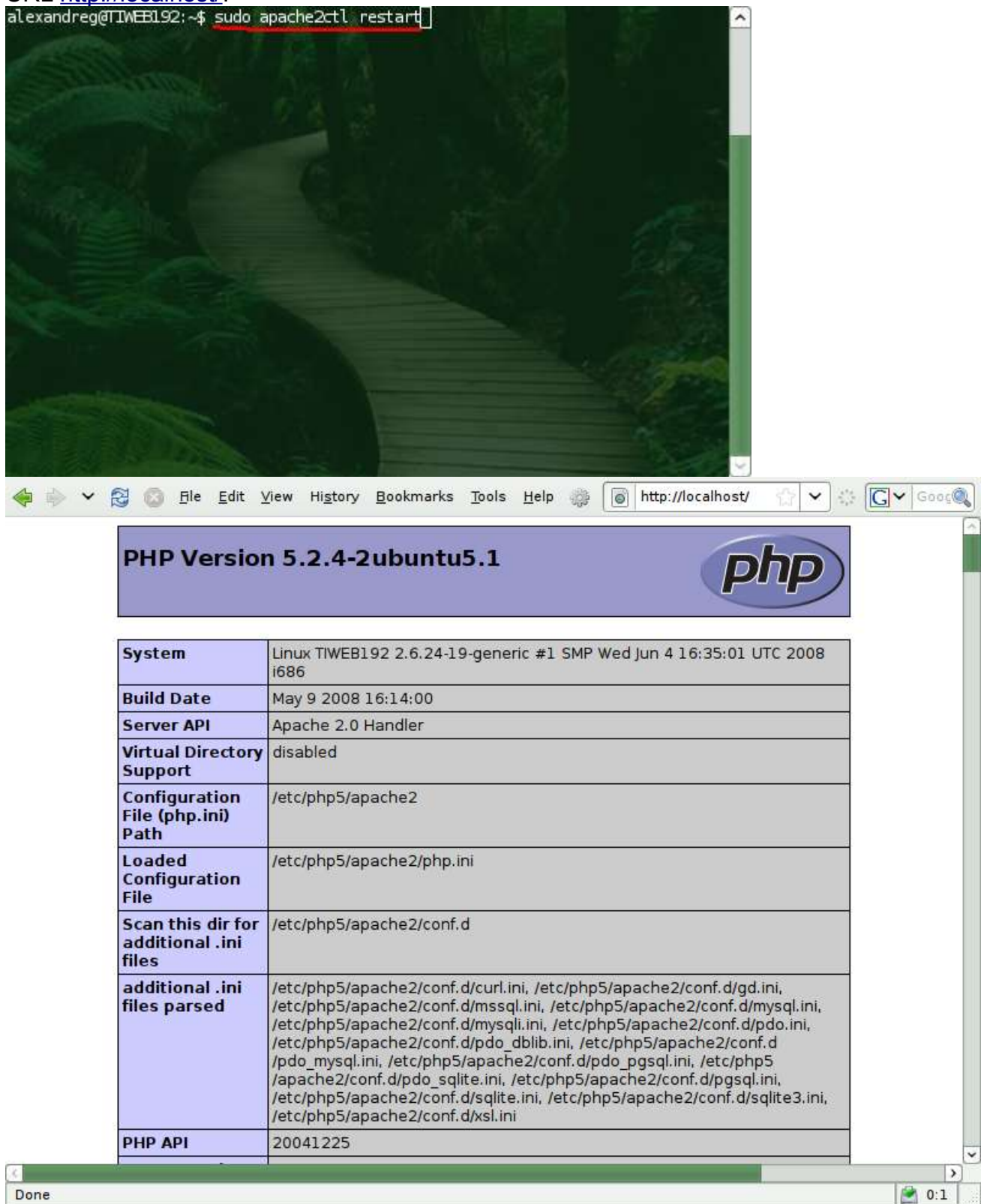
Ao acessar <http://localhost/>, você deverá ver uma tela de informações do PHP. Geralmente isso não ocorre, e o apache envia o arquivo para download.



Done

0:1

Se isso ocorrer, basta reiniciarmos o apache digitando no terminal `sudo apache2ctl restart` (para abrir o terminal, aperte Alt+F2 e digite `gnome-terminal`). Isso recarregará as configurações e tudo deverá funcionar normalmente, exibindo o PHP Info ao acessar a URL <http://localhost/>.



The screenshot shows a terminal window at the top with the command `alexandreg@TIWEB192:~$ sudo apache2ctl restart` entered. Below the terminal is a web browser window displaying the PHP information page for version 5.2.4-2ubuntu5.1. The page has a purple header with the PHP logo. Below the header is a table with system and configuration details.

System	Linux TIWEB192 2.6.24-19-generic #1 SMP Wed Jun 4 16:35:01 UTC 2008 i686
Build Date	May 9 2008 16:14:00
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
additional .ini files parsed	/etc/php5/apache2/conf.d/curl.ini, /etc/php5/apache2/conf.d/gd.ini, /etc/php5/apache2/conf.d/mssql.ini, /etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/mysqli.ini, /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d/pdo_dblib.ini, /etc/php5/apache2/conf.d/pdo_mysql.ini, /etc/php5/apache2/conf.d/pdo_pgsql.ini, /etc/php5/apache2/conf.d/pdo_sqlite.ini, /etc/php5/apache2/conf.d/pgsql.ini, /etc/php5/apache2/conf.d/sqlite.ini, /etc/php5/apache2/conf.d/sqlite3.ini, /etc/php5/apache2/conf.d/xsl.ini
PHP API	20041225

At the bottom of the browser window, a status bar shows "Done" and a timer at "0:1".

## Testando o MySQL

Só nos resta testar o MySQL. Para isso abriremos um terminal e digitaremos o comando `mysql -u root -p`, que conecta ao banco de dados local utilizando o administrador (usuário `root`). Esse comando pedirá uma senha, basta digitar aquela que você escolheu na configuração do MySQL lá em cima.

```
alexandreg@TIWEB192:~$ mysql -u root -p
```

Se tudo correr bem, o prompt do monitor do MySQL estará disponível para digitar instruções.

```
alexandreg@TIWEB192:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.0.51a-3ubuntu5.1 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> 
```

Parabéns! Tudo funcionou corretamente?

### Considerações Finais:

Este material é um apanhado de tutoriais destinados a serem executados em sala de aula e como exercícios extras, na disciplina de Gestão de Sistemas Operacionais II.

Não recomendo que este material seja impresso, pelo bem da natureza, somente salvo em um pen drive e utilizado como apoio durante as aulas e para estudo.

Os endereços dos tutoriais, seus autores e referências bibliográficas seguem citados no decorrer da apostila.