INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RN  
DIRETORIA ACADÊMICA DE GESTÃO E INFORMÁTICA  
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

*Ao professor Leonardo Minora  
para a disciplina de Sistemas Operacionais*

**RELATÓRIO DE DESENVOLVIMENTO DO SISTEMA OPERACIONAL LINUX, BASEADO NO LIVRO LINUX FROM SCRATCH (LFS).**

*Suzyanne de O. Queiroz*

*07/02/2014*

**INTRODUÇÃO**

Para composição parcial da nota da disciplina de Sistemas Operacionais foi proposta a compilação de um sistema operacional Linux. Foi usado como ferramenta de ajuda e estudo o tutorial do Linux From Scrach, disponível em : <http://www.linuxfromscratch.org/lfs/view/stable/>. O livro fornece instruções necessárias para a criação do sistema Linux, enfocando as vantagens de utilizar tal sistema, tais como a disposição dos diretórios, a instalação dos scripts e questões de segurança. Em suma, o tutorial pretende dar ao usuário a liberdade do arranjo do sistema final, compilando, do zero, todas as partes do projeto a partir de códigos-fonte. Antes de iniciado o tutorial, a leitura do Guia Foca Linux fez-se necessária, a fim esclarecer comandos e parâmetros básicos para o entendimento do processo de criação do sistema, para usuários que não possuírem domínio prévio do shell. Foi recomendada pelo professor, a utilização de uma máquina virtualizada como sistema anfitrião. Tal medida traz vantagens no salvamento e recuperação de etapas desenvolvidas e não afetar os recursos de hardware da máquina hospedeira.

**DESENVOLVIMENTO**

Para hospedar o novo sistema operacional, foi virtualizado, em primeira instância, o Ubuntu 12.04 32 bits. O KERNEL do sistema anfitrião foi compilado com gcc e é compatível com as exigências. A verificação foi feita, em shell, através do comando:

**cat /proc/version**

Para hospedar o LFS, foi necessária preparação de uma partição dedicada com tamanho maior que o sistema final já compilado, devido à necessidade de espaço adicional durante a compilação dos pacotes.

Na primeira tentativa de particionamento, através do comando fdisk, a tabela de particões do sistema anfitrião foi alterada, incluindo a referência para partição de boot, acarretando em quebra do sistema após sua reinicialização. Para resolver o problema, foi adicionado um disco virtual VDI de 40 Giga e este, particionado em dois: uma para a instalação do sistema e outra para o swap e, em seguida, suas respectivas formatações.

Para ver se o sistema host possuia todas as versões apropriadas, e a capacidade de compilar programas, um script de shell foi criado e rodado. As versões do bison, gawk, g++, patch e makeinfo (4.13) estavam pendentes e foram devidamente instaladas.

O ponto de montagem do sistema foi definido em **/mnt/lfs**. A variável de sistema $LFS foi configurada para apontar a este diretório e utilizada durante todo o processo de montagem:

**Export LFS=/mnt/lfs**

Em seguida, com acesso *root*, o diretório de trabalho *sources* foi definido e a pasta modificada a partir do comando chmod, a fim de torná-la "writable" e Sticky. Um diretório Sticky, mesmo provendo permissão de escrita a vários usuários, somente pode ter arquivos deletados pelo usuário que o criou:

**mkdir -v $LFS/sources**

**chmod -v a+wt $LFS/sources**

Packages e patches foram baixados utilizando o comando wget. Este deve baixar arquivos de uma rede de forma não iterativa, utilizando protocolos http, https, ftp e proxies http. O comando é geralmente associado à função de download recursivo.

**wget –i http://www.linuxfromscratch.org/lfs/view/stable/wget-list -P $LFS/sources**

Na verificação dos packages baixados, observou-se a ausência do package Shadow (4.1.5.1), posteriormente localizado na internet e adicionado ao diretório *sources*. Também utilizado o recurso do comando wget, foram baixados os patches para correção de possíveis erros.

Todos os programas a serem compilados no sistema temporário devem ser instalados em *$LFS/tools*. Foi então criado o diretório, a partir do comando *mkdir*.

Também em modo *root,* foi criado um link simbólico direcionando o termo /tools para $LFS/tools, a fim de encaminhar corretamente ao local de compilação:

**ln -sv $LFS/tools /tools**

Para evitar deslizes ou danos irrecuperáveis através das permissões *root*, foi recomendado montar os pacotes com um usuário sem tais privilégios. Os comandos a seguir foram executados para, respectivamente, adicionar um grupo de *users*, adicionar um usuário chamado *lfs2,* vinculando *bash* como *shell* padrão e configuração de senha:

**groupadd lfs**

**useradd -s /bin/bash -g lfs -m -k /dev/null lfs**

**passwd lfs**

Para que o novo usuário tivesse acesso às pastas necessárias para a compilação dos pacotes, o comando *chown* foi aplicado aos diretórios *$LFS/tools* e *$LFS/sources*. Nesta etapa, tivemos dificuldade de executar o comando, por problemas com o link simbólico executado incorretamente. Segue a saída do comando:

**failed to change ownership of `/mnt/lfs/sources' to lfs**

**chown: cannot access `/mnt/lfs/tools': No such file or directory**

Os links simbólicos foram refeitos e o comando reexecutado com sucesso:

**changed ownership of `/mnt/lfs/sources' from root to lfs2**

**changed ownership of `/mnt/lfs/tools from root to lfs2**

Antes da construção do sistema temporário, a configuração do ambiente foi feita, como usuário *lfs2*, impedindo que alguma variável não desejada do sistema hospedeiro tenha acesso indevido ao ambiente de compilação. A seguir, o arquivo *.bashrc* foi criado para ser lido no lugar do *.bash\_profile.* O novo arquivo desativa a função **hash** do **bash**, obrigando o **shell** a sempre procurar o **PATH** quando um programa for rodar. Também é criada uma máscara de usuário que dá permissões de escrita apenas ao dono, mas permite leitura e execução por qualquer um. A variável **LC\_ALL** padroniza o ambiente nas definições do POSIX e evita problemas de saída e retorno ao ambiente. Por fim, colocar a pasta **/tools/bin** no inicio do **PATH**, combinada com a desativação da **hash**, faz com que os programas a serem compilados sejam imediatamente acessados pelo **shell** após a instalação.

Durante a etapa de compilação do sistema temporário, encontramos algumas dificuldades em relação à configuração dos links simbólicos que deveriam estar seguramente definidos. Como não encontramos, no livro, definição anterior de tais links, executamos o comando **ln** para as referências citadas, mas isso gerou redundância e loops infinitos, prejudicando o andamento da etapa. Apenas o pacote binutils-2.23.2 foi compilado, com auxilio do professor. A compilação do pacote seguinte, o **gcc**, não pôde ser concluída por dificuldade na interpretação das saídas de erro, tanto no **shell** quanto nos arquivos **config.log.**

**ANEXOS**

**Anexo 1 – Cópia do arquivo ‘logfile’ contendo saídas de comandos importantes**

**# Comentários serão precedidos com '#'**

**# O restante do conteúdo foi gerado por redirecionamento de saída**

**# 13:12:2013**

**ln: failed to create symbolic link `/tools': File exists**

**# Havia uma pasta já criada com o nome de referencia para o**

**# link a ser criado. Pasta removida e comando refeito:**

**`/tools' -> `/mnt/lfs/tools'**

**chown: cannot access `/mnt/lfs/tools': No such file or directory**

**failed to change ownership of `/mnt/lfs/tools' to lfs**

**chown: cannot access `/mnt/lfs/sources': No such file or directory**

**failed to change ownership of `/mnt/lfs/sources' to lfs**

**chown: cannot access `/mnt/lfs/tools': No such file or directory**

**failed to change ownership of `/mnt/lfs/tools' to lfs**

**ln: failed to create symbolic link `/tools': File exists**

**`/tools' -> `/tools'**

**chown: cannot dereference `/tools': Too many levels of symbolic links**

**failed to change ownership of `/tools' from 4294967295 to lfs2**

**chown: cannot access `/sources': No such file or directory**

**failed to change ownership of `/sources' to lfs2**

**chown: cannot access `/mnt/lfs/tools': No such file or directory**

**failed to change ownership of `/mnt/lfs/tools' to lfs2**

**ownership of `/mnt/lfs/tools' retained as lfs2**

**changed ownership of `/mnt/lfs/sources' from root to lfs2**

**06/01/2014**

**mkdir -p -- /tools /tools**

**mkdir: cannot create directory `/tools': File exists**

**mkdir: cannot create directory `/tools': File exists**

**make[1]: \*\*\* [installdirs] Error 1**

**make[1]: Leaving directory `/mnt/lfs/sources/binutils-build'**

**make: \*\*\* [install] Error 206/01/2014**

**mkdir -p -- /tools /tools**

**mkdir: cannot create directory `/tools': File exists**

**mkdir: cannot create directory `/tools': File exists**

**make[1]: \*\*\* [installdirs] Error 1**

**make[1]: Leaving directory `/mnt/lfs/sources/binutils-build'**

**make: \*\*\* [install] Error 2**

**Anexo 2 – Cópia do arquivo de rascunho para acompanhamento das etapas do projeto**

**Relatorio de acompanhamento da instalação do sistema operacional Linux no Daruma**

**12/11**

**Início da leitura do livro LINUX FROM SCRATCH (VERSÃO EM PORTUGUES)**

**Fonte de apoio: http://goo.gl/7pHA8y**

**Tentativa de emular o sistema operacional montado LFS realizada com sucesso.**

**13/11**

**Instalação do SO UBUNTU na máquina virtual para prosseguir com os passos do LFS Book.**

**O KERNEL do sistema anfitrião foi compilado com gcc e é compatível com as exigências.**

**25/11**

**Criação de arquivo de texto utilizando o "cat"**

**cat < teste.txt << "EOF"**

**Trecho relevante:**

**"O programa chroot (change root) é usado para entrar em um ambiente virtual e inicializar um novo shell cujo o diretório de raiz será definido na partição do LFS. Isto é muito similar a reinicializar e a instruir o kernel para montar a partição LFS como a partição root. O sistema não reinicializa realmente, mas faz um chroot porque criar um sistema inicializável requer o trabalho adicional que não é necessário neste momento. A vantagem principal de “chrooting” é permitir o uso do sistema anfitrião enquanto o LFS estiver sendo configurado. Enquanto espera a compilação de algum pacote terminar, o usuário pode abrir um console virtual diferente (VC) ou o desktop X e continuar usando seu computador normalmente."**

**-------------------------**

**Criando uma nova partição:**

**Trecho de estudo sobre particionamento de disco**

**Consulta de partições do sistema através do comando**

**sudo fdisk -l**

**Em seguida, configuração de novas partições através de**

**sudo cfdisk**

**Duas novas particões foram criadas, dividindo a memória em partes iguais, sendo uma delas bootavel.**

**Observação: Após alteração das partições do da Virtul Machine (VM), a linha de comando segue apresentando erros de inicialização e não reconhece o vinculo do sudoers para autenticação do comando sudo.**

**-----------------------**

**Maquina virtual quebrada. Realização do processo de instalação de novo SO virtualizado (nome: tads2).**

**-----------------------**

**26/11**

**Continuação de estudos sobre particionamento.**

**Criação de novo disco virtualizado pela Virtualbox com tamanho de 40G**

**divisão do novo disco em 4 partições através do**

**fdisk /dev/sdb**

**27/11**

**2.3 Formatação de partições e arquivo de sistema**

**Partição /dev/sdb1 formatada em ext4,**

**criação e formatação de swap de 2GB em dev/sdb5**

**mkswap /dev/sdb5**

**Obs.: sdb2 deletada**

**2.4 Montando nova partição**

**escolhido o ponto de montagem /mnt/lfs para exportar a variável de ambiente LFS**

**export LFS=/mnt/lfs**

**criado o ponto de montagem do arquivo de sistema**

**mkdir -pv $lfs**

**e montado em**

**mount -v -t ext4 /dev/sdb1 $LFS**

**---**

**02/12/2013**

**Criação de diretório de trabalho a partir do comando:**

**mkdir -v $LFS/sources**

**Em seguida, pasta modificada a partir do comando chmod, a fim de torná-la "writable" e Sticky. Um diretório Sticky, mesmo provendo permissão de escrita a vários usuários, somente pode ter arquivos deletados pelo usuário que o criou.**

**chmod -v a+wt $LFS/sources**

**Packages e patches baixados utilizando o comando wget. Este fornece baixa arquivos de uma rede de forma não iterativa, utilizando protocolos http, https, ftp e proxies http. O comando é geralmente associado à função de download recursivo.**

**Devido a problemas com a máquina onde a virtualização estava sendo rodada, optou-se pela troca de sistema operacional, no entando a maquina virtual criada não foi recuperada**

**----**

**12/12/2013**

**Nova máquina virtual criada (Ubuntu 12.04) e passos efetuados anteriormente refeitos.**

**seguindo para a o capítulo 4: Final Preparations.**

**----**

**16/12/2013**

**criação do diretório tools em $LFS**

**mkdir -v $LFS/tools**

**através do comando ln, um link simbólico no sistema principal para a pasta recém-criada tools será feito**

**Criação de novo grupo de usuários no qual será criado um usuário sem privliégios root.**

**groupadd lfs2**

**useradd -s /bin/bash -g lfs2 -m -k /dev/null lfs2**

**Criação de novo usuário 'lfs2' (no grupo lfs2) que será usado para a compilação dos pacotes, por não ter permissões root, evitando danificar o pc.**

**o usuário lfs2 tem acesso irrestrito garantido às pastas tools e sources através do comando:**

**chown -v lfs2 $LFS/tools**

**chown -v lfs2 $LFS/sources**

**4.4 Configurando o ambiente**

**definição de dois arquivos base para o shell do interpretador de comandos 'bash':**

**cat > ~/.bash\_profile << "EOF"**

**exec env -i HOME=$HOME TERM=$TERM PS1='\u:\w\$ ' /bin/bash**

**EOF**

**cat > ~/.bashrc << "EOF"**

**set +h**

**umask 022**

**LFS=/mnt/lfs**

**LC\_ALL=POSIX**

**LFS\_TGT=$(uname -m)-lfs-linux-gnu**

**PATH=/tools/bin:/bin:/usr/bin**

**export LFS LC\_ALL LFS\_TGT PATH**

**EOF**

**Para finalizar a preparação, o comando a seguir é executado para informar ao shell em andamento as diretrizes passadas no arquivo de parâmetro:**

**source ~/.bash\_profile**

**20/12/2013**

**Na etapa 5.3, detalhes importantes foram enfatizados como a criação de links simbolicos para bash, gawk e bison.**

**A definição dos links simbolicos causou um loop que impede a procedimentos básicos como a função 'man'.**

**A máquina será descartada e Snapshot do dia 16/12 (etapa 4.4) será reutilizado.**

**2 BIMESTRE**

**10/02/2014 a 27/02/2014**

**Foi preciso voltar para o capítulo 5 para entender melhor como seria a construção do sistema provisório LFS, a proposta desse capítulo é permitir um ambiente de trabalho mais conveniente para o usuário de como um ambiente mínimo do LFS faria.**

**O próximo passo é compilar os pacotes para construir esse sistema provisório e esses arquivos compilados serão instalados no LFS/tools $ diretório para mantê-los do diretório do sistema hospedeiro que será construído mais na frente.**

**O primeiro pacote a ser compilado será o Binutils porque o configure de execuções de ambos GCC e Glibc vão realizar vários testes de recurso no assembler e linker para determinar quais recursos de software serão ativados e desativados.**

**Então foi verificado se a variável de ambiente está configurada corretamente, com o comando :**

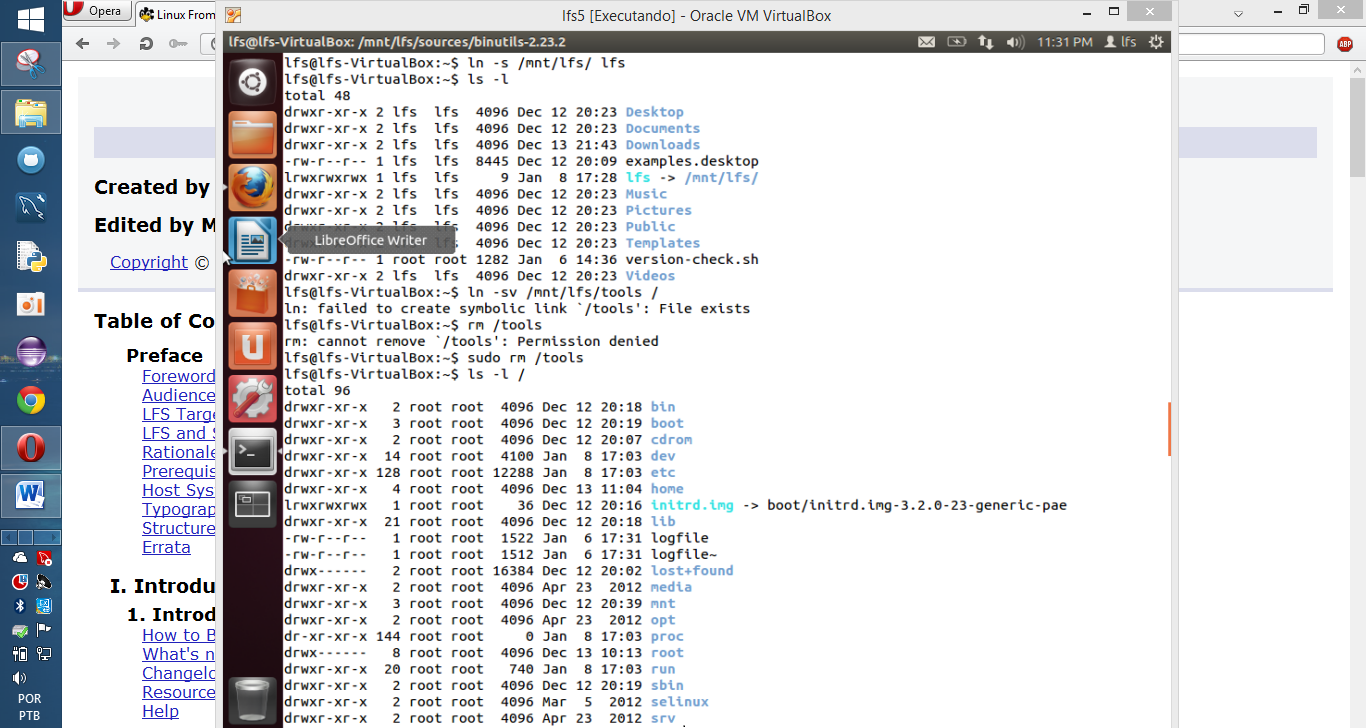
**Echo $ LFS**

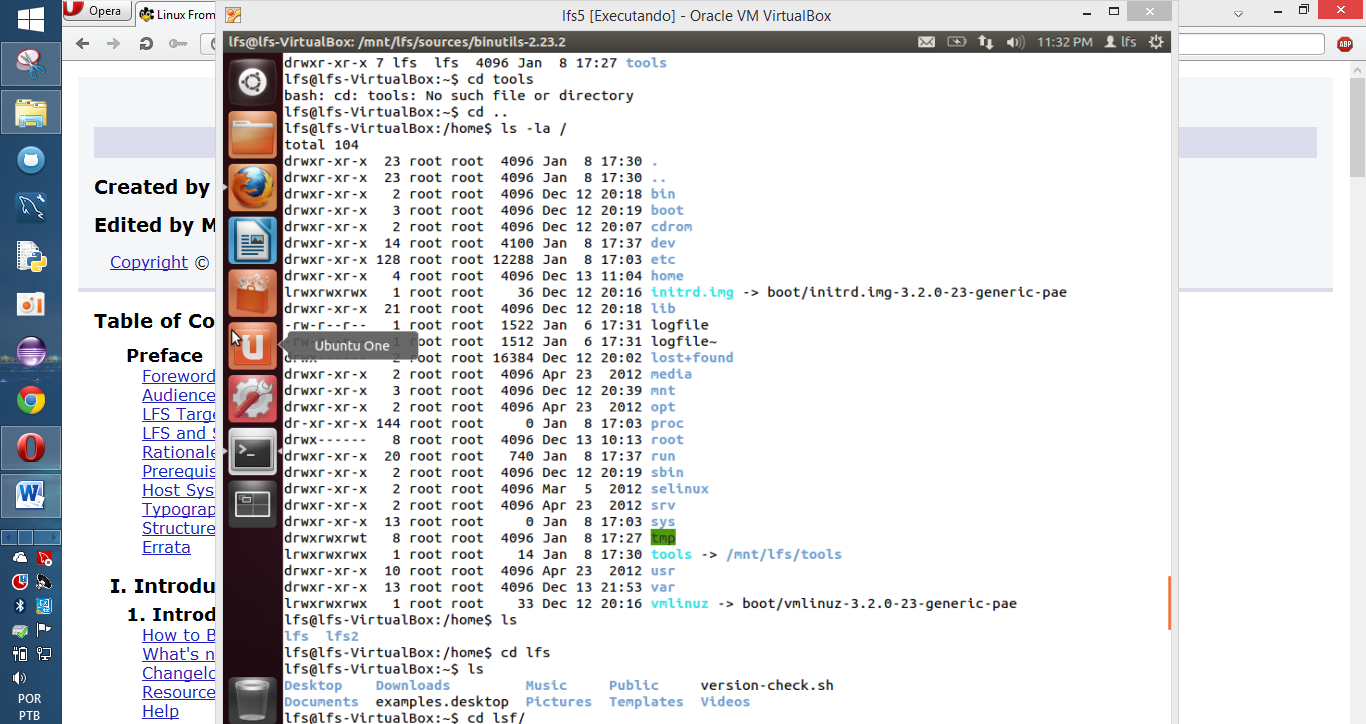
**E o resultado foi: /mnt/lfs**

**Segundo passo importante foi incluir os links simbólicos corretamente,**

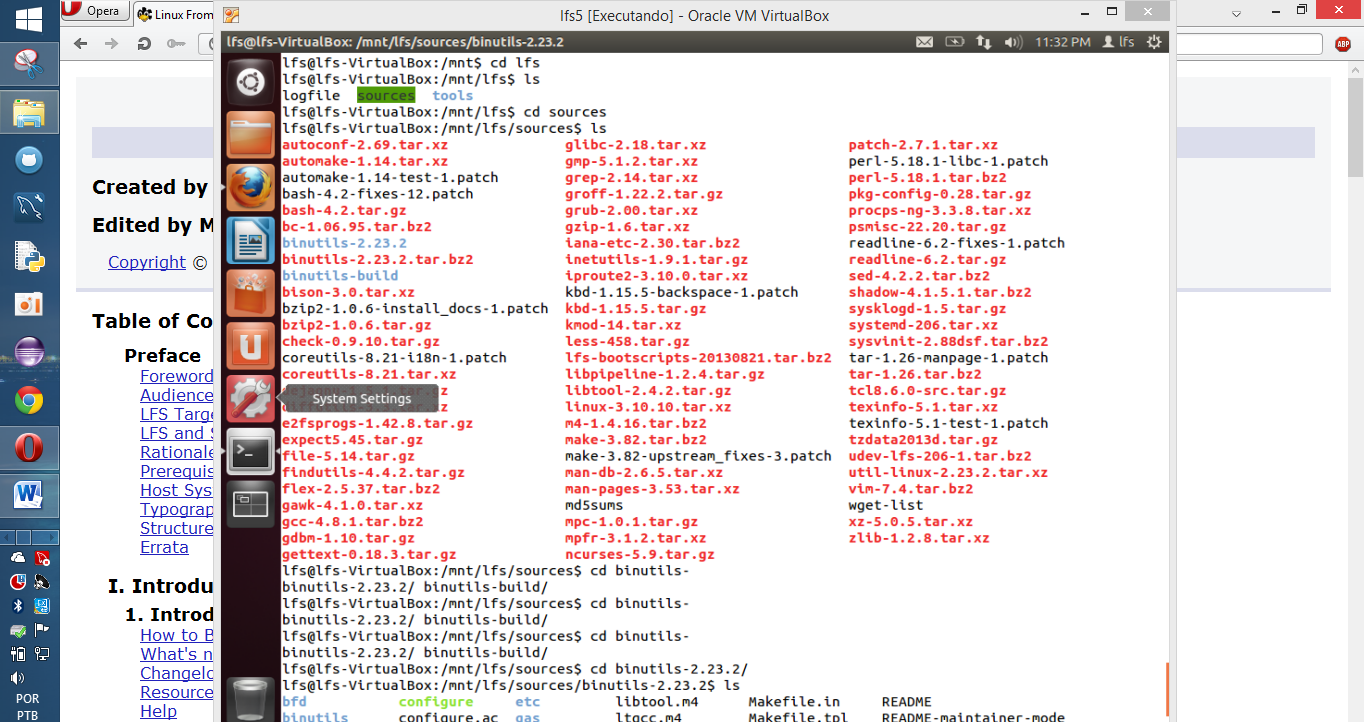
**/usr/bin/awk é um link simbólico para gawk**

**/usr/bin/yacc é um link simbólico para bison**

****

****

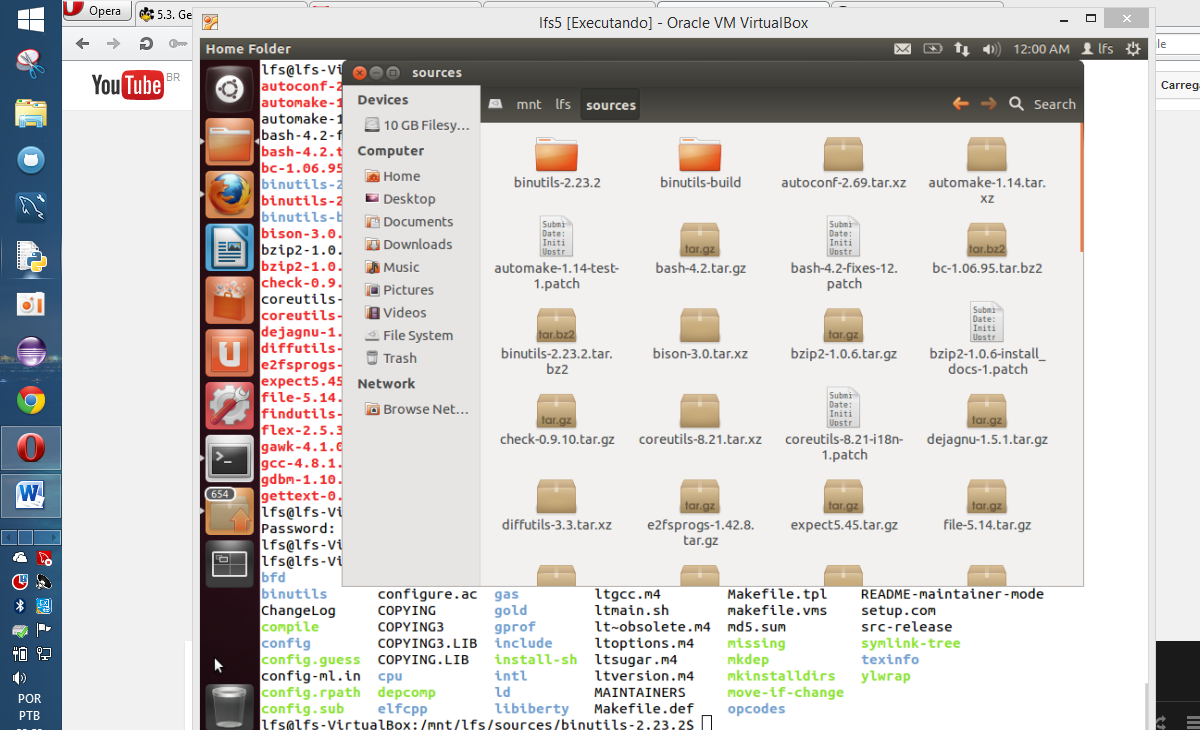
Outro passo proposto no Linux From Scratch foi colocar todas os sources e patches em um diretório que será acessível a partir do ambiente chroot, ou seja colocar tudo em /mnt/LFS/sources / e não em /mns/LFS/tools/.



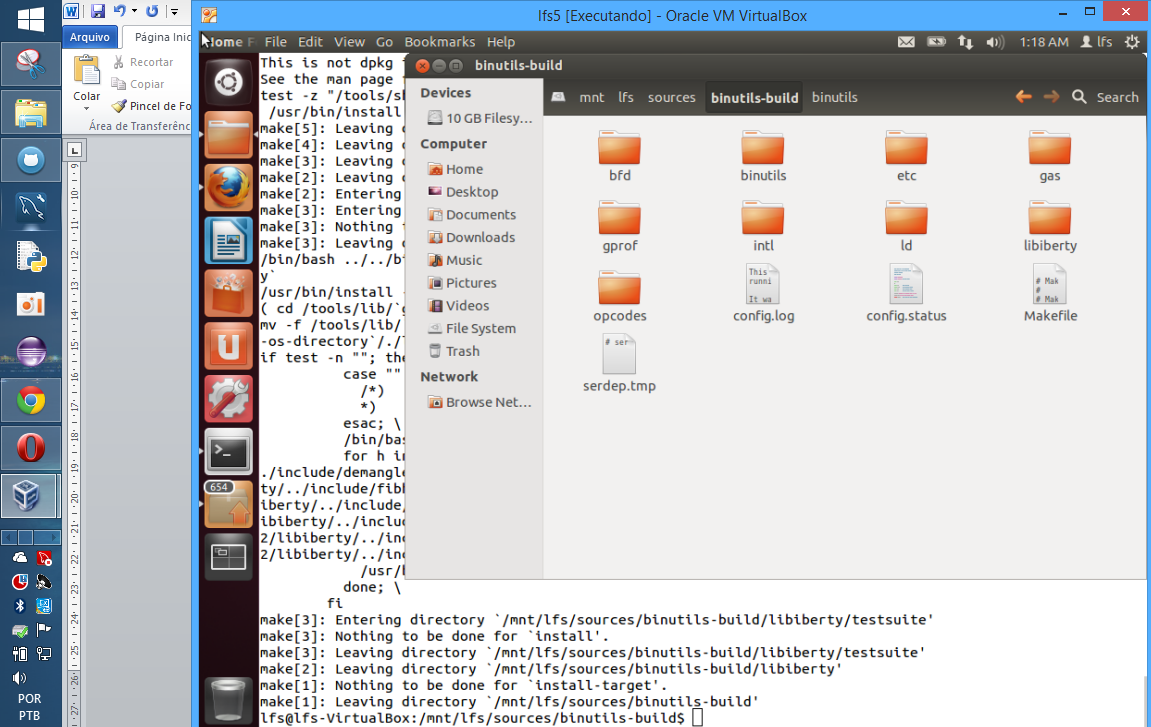
Na compilação do Binutils 2.24 foi observado tuas informações importantes tais como : o tempo de compilação aproximada que é de 1SBU e o espaço em disco necessário 404 MB, importante entender essas informações para já ir entendendo os tamanhos dos pacotes para saber o que vamos realmente precisar no LFS final onde teremos que ter um Sistema Operacional de 512 MB.

O primeiro passo para compilar o Binutils foi extrair o pacote, onde extrai manualmente na pasta e por comando crei um diretório de compilação dedicado, através do comando :

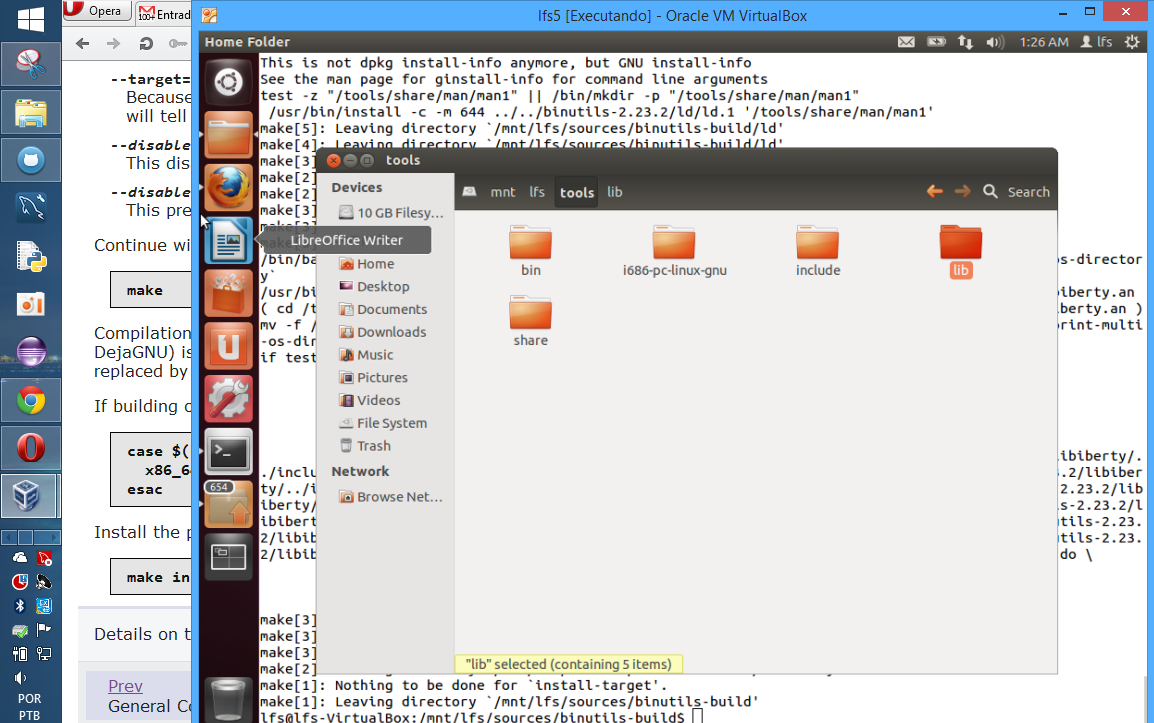
mkdir -v ../binutils-build



Depois foi dado o comando make install para a instalação do pacote. As duas figuras a baixo mostra como ficou depois do make isntall e foi observado que ocorreu mudanças nas pastas Sources e Tools , arquivos novos foram gerados.



**Pasta SOURCES**



**Pasta TOOLS**

O último passo referente ao Binutils foi remover o pacote que foi extraído [Binutils – Build] com o comando rm –rf binutils-build , a figura a baixo mostra com detalhes.

