

Pontifícia Universidade Católica de Campinas Faculdade de Engenharia de Computação - FECOMP

Sistemas Operacionais B – Relatório Atividade 1

Beatriz Morelatto Lorente RA: 18071597

Cesar Marrote Manzano RA: 18051755

Fabricio Silva Cardoso RA: 18023481

Pedro Ignácio Trevisan RA: 18016568

**Sumário**

**1.Introdução.......................................................................................................3**

**2.Principais passos da compilação de um kernel Linux...............................4**

**3.Conclusão.......................................................................................................7**

**Introdução**

O experimento tem como objetivo familiarizar-se com todas as etapas de compilação de um kernel. Foi experimentado todo o processo de instalação, configuração e compilação do kernel. Foi necessário compreender cada fase para obtermos o resultado desejado, ou seja, com instalar um kernel, o que podemos desativar em um kernel Linux e como fazer sua compilação. Ao final do experimento é esperado que a equipe obtivesse um kernel que pudesse ser compilado rapidamente, com no máximo 30 minutos.

**Principais passos da compilação de um kernel Linux**

Passo a passo para a compilação de um kernel

Para a compilação de um kernel é necessário ter em mente que dados da sua máquina podem ser corrompidos, sua máquina pode não rodar mais, uma vez que ocorreu algum erro no kernel. Portanto é recomendado o uso de uma máquina virtual, ao invés de instalar o sistema Linux em sua máquina. Para o experimento foi utilizado o Ubuntu 16.04.

Para iniciarmos o processo de compilação é necessário baixar uma versão do kernel. Para isso, foi usado o comando *‘sudo apt-cache search linux-source’*, no terminal do ubuntu, como mostrado na imagem abaixo.

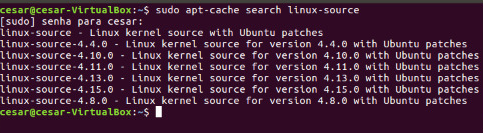


Figura 1 - Print mostrando o comando para procurar uma versão do kernel linux

Ao executar o comando o terminal mostra várias opções disponíveis de kernel para a instalação. Para o experimento foi utilizado a versão 4.15.0. Para o arquivo ser baixado basta digitar o comando *‘sudo apt-get install linux-source-4.15.0’* (se quiser baixar outra versão apenas troque os números da versão do linux).

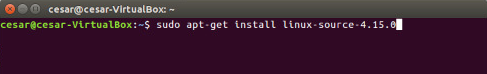
****

Figura 2 - Print mostrando o comando para instalar a versão desejada do kernel

A primeira parte para a compilaçao do kernel está feita. O downloads do arquivo se encontra na pasta *‘usr/src’*. Basta navegar pelo terminal para acessá-la. Ao chegar na pasta vamos encontrar uma variedade de arquivos, como mostrado abaixo.

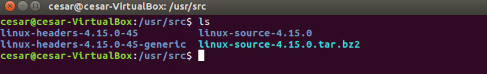


Figura 3 - Print mostando os arquivos da pasta 'usr/src'

O arquivo desejado é o *‘linux-source-4.15.0.tar.bz2’.* É recomendado que voce copie esse arquivo para uma pasta separada, assim evitando qualquer tipo de conflito com os outros arquivos da pasta atual. Para copiar um aruquivo digite o comando *‘cp linux-source-4.15.0.tar.bz2 ‘caminho-da-pasta-desejada’’.*

Com isso feito é necessário descompactar o arquivo para podermos manipular o arquivo. Para descompactar digite o comando *‘tar xjpf linux-source-4.15.0.tar.bz2’.* Espere a descompactação do arquivo e você terá um resultado como na imagem abaixo.

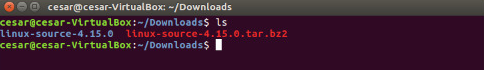


Figura 4 - Print mostrando arquivo descompactado

Antes de prosseguir será necessário instalar algumas bibliotecas para que a compilação possa ser feita de maneira correta. Abaixo estão as bibliotecas necessárias e seus respectivos comando para instalação.

* libncurses5: *‘sudo apt-get install libncurses5-dev’*
* buld-essential: *‘sudo apt-get install build-essential’*
* libssl: *‘apt install libssl-dev -y’*

Com as bibliotecas instaladas podemos dar continuidade ao processo de compilação.

Próximo passo é virar root da máquina. Para isso digite o comando *‘sudo su’*. Virando root da máquina os passos finais da compilação poderão ser iniciados de fato. Primeiro comando a se digitar é o *‘make menuconfig’*. Com esse comanda uma tela se abrirá com as configurações do linux. Você poderá navegar nesse menu desativando módulos que são desnecessários em sua máquina, diminuindo assim o tempo de compilação total.

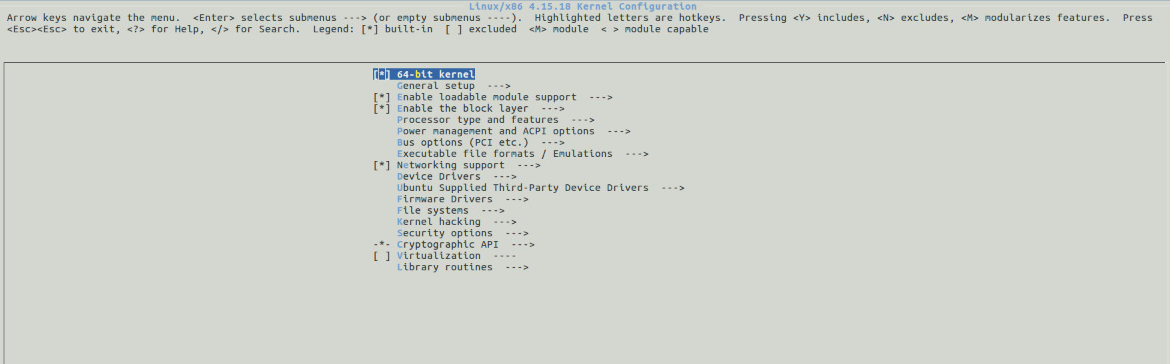


Figura 5 - Print da tela mostrando o menu de configuração do kernel

No topo do menu há pequenas instruções, indicando como navegar pelo menu, como ativar/desativar módulos, etc. O importante nessa fase é analisar o que é e o que não é essencial para o kernel. A função da grande maioria dos módulos é desconhecida, portanto há o risco de algo fundamental ser desativado. Para que esse erro não comprometesse o desenvolvimento do experimento, foram sendo criadas cópias da máquina virtual, conforme obtivemos versões de kernel que funcionassem.

Após a desativação dos módulos, é necessário compilar o kernel de fato com suas novar configurações. Para que isso ocorra digite o comando ‘*make’*no terminal. Caso sua máquina virtual esteja habilitado para rodar com mais de uma CPU, você pode usar o comando *‘make -jN’,* sendo que ‘N’ é o número de CPUs que você tem disponível para a máquina virtual.Esse comando, como citado, é responsável pela compilação, criando um binário do kernel.

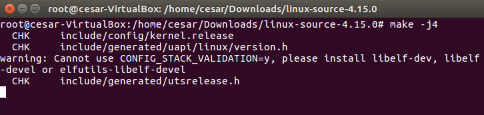


Figura 6 - Print mostrando o comando ‘make’

Após se compilar o kernel é necesssário fazer uma instalação dinâmica dos módulos do kernel. Para isso é usado o comando *‘make modules\_install’.*



Figura 7 - Print mostrando o comando 'make modules\_install'

Após este comando basta apenas instalar o kernel em sua máquina com o comando *‘make install’*.

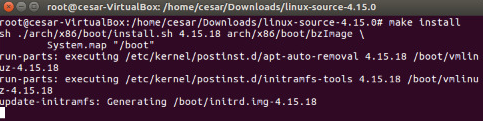


Figura 8 - Print mostrando o comando 'make install'

Após a instalaçao ser concluída basta reiniciar sua máquina com o comando *‘reboot’.*

Possíveis erros durante a compilação do kernel

* *Makefile: 1086: recipe for target .'drivers' failed*: quando a compilação do kernel é finalizada é possível se depar com esse erro, no qual alguns drivers retornavam erro. O próprio ubuntu nos dá a solução, dizendo que é necessário instalar a lib bison. Para instalar a biblioteca, usa-se o comando ‘*sudo apt-get install bison flex’.* Anteriormente citamos algumas bibliotecas que precisam ser instaladas para a compilação do kernel. Há casos em que esta biblioteca não precisa ser instalada, portanto não está citada anetriormente.
* *Make: \*\*\* [menuconfig] Error 2:* Esse erro só irá ocorrer caso não se instale a biblioteca libssl.
* *Make: \*\*\* No rule to make target 'menuconfig'. Stop:* Esse erro ocorre ao tentar abrir o menu para ter acesso ao kernel para modificá-lo. Ocorre caso você não esteja como root da máquina ou esteja tentando rodar o comando *‘make menuconfig’* na pasta errada.
* *Kernel Panic - Not syncing: out of memory and no killable process:* Esse erro ocorre na reinicialização (reboot) da máquina após o kernel ser compilado. Para este erro de kernel panic, há a necessidade de adicionar mais memória à sua máquina virtual.

**Conclusão**

A maior parte dos módulos de um kernel são desconhecidos pela maioria das pessoas, até mesmo para aquelas que trabalham no setor de computação. Portanto, para a compilação de um kernel enxuto era necessário se arriscar e desativar módulos que possivelmente são essenciais para o kernel. Esse processo leva a diversos erros, que são de extrema importância para o entendimento do que poderia e o que não poderia ser feito dentro de um kernel. Com esses e outros erros, foi possível compreender todos os passos para compilação de um kernel de forma clara.