

Documentación

LINUX FROM SCRATCH (LFS)

SEGUNDO SEMESTRE 2021

SISTEMAS OPERATIVOS 1

Ramon Araujo Y09157

Araceli Olmedo Y08571

Preparándose para la construcción

Esta es la primera etapa de la construcción del LFS, en la cual explicaremos brevemente los que realizamos desde el capítulo 1 al 4 para prepararnos para la construcción. Esta primera etapa lo pudimos concluir con éxito, sin problema alguno.

Nombre de usuario: ara

Contraseña: 12345

Disco SDA 25GB

Disco SDB 25 GB

RAM 8GB

Lo primero que realizamos fue la instalación de una lista de paquetes requeridas por el host con una versión mínima indicada en el libro en la sección 2.2, como lo vemos a continuación:

- **Bash-3.2**
- **Binutils-2.25**
- **Bison-2.7**
- **Bzip2-1.0.4**
- **Coreutils-6.9**
- **Diffutils-2.8.1**
- **Findutils-4.2.31**
- **Gawk-4.0.1**
- **GCC-6.2**
- **Glibc-2.11**
- **Grep-2.5.1a**
- **Gzip-1.3.12**
- **Kernel de Linux-3.2**
- **M4-1.4.10**
- **Hacer-4.0**
- **Parche-2.5.4**
- **Perl-5.8.8**
- **Python-3.4**
- **Sed-4.1.5**
- **Alquitrán-1.22**
- **Texinfo-4.7**
- **Xz-5.0.0**

Estos paquetes fueron instalados con el comando:

Sudo apt-get install <Nombre del Paquete>

Luego, para una verificación de que el sistema host tiene todas las versiones apropiadas y la capacidad de compilar programas ejecutamos un script llamado `version-check.sh`

Siguiendo los pasos del libro, creamos dos particiones del disco `sdb`. En la partición `sdb1` de 20GB se encuentra nuestro sistema LFS, la partición `sdb2` de 5GB fue destinada para el swap, el sistema de archivo utilizado en la partición es el `ext4`, especificado en los requerimientos del trabajo.

A continuación creamos la variable `LFS`, en la cual cargamos el nombre del directorio donde construimos el sistema LFS, esto lo hacemos con el comando:

```
Export LFS=/mnt/lfs
```

En el capítulo 3, creamos un script llamado **wget-list** donde se encuentran todos los links de los paquetes y parches descargados, para ser instalados posteriormente. Con el siguiente commando:

```
Wget --input-file=wget-list --continue --directory-prefix=$LFS/sources
```

Llamamos a este script para realizar la descarga, indicando que se deben instalar en el directorio `/mnt/lfs/sources`. Para verificar que todos los paquetes correctos estén disponibles antes de continuar utilizamos `md5sums` para comparar con nuestros paquetes instalados, el cual es un script con los códigos y nombres de los distintos paquetes.

Para no dañar o destruir el sistema (utilizando el usuario `root`), creamos un usuario sin privilegios llamado `lfs`, al cual otorgamos acceso completo a todos los directorios bajo el dominio de `$LFS` haciendo que `lfs` sea propietario.

```
chown -v lfs $LFS/{usr{,/}*},lib,var,etc,bin,sbin,tools}
case $(uname -m) in
  x86_64) chown -v lfs $LFS/lib64 ;;
esac
```

En este nuevo usuario volvemos a hacer las configuraciones correspondientes, como por ejemplo volver a crear la variable `LFS`, el cual lo guardamos en `.bashrc` para que se ejecute al iniciar el sistema.

Creación de la cadena de herramientas cruzada LFS y herramientas temporales

En esta segunda etapa comentaremos sobre los procedimientos realizados desde el capítulo 5 al 7, como así también sobre los errores que se fueron presentando y sus respectivas soluciones.

En esta sección comenzamos instalando el paquete Binutils, el cual tiene una duración de 1SBU y nos sirvió de base para hacer la conversión de SBU/min, llegando a la conclusión de que 1SBU= 3minutos, el cual fue mucha ayuda para luego determinar cuántos minutos nos tomaría la instalación de los demás paquetes.

Los pasos para la instalación de los paquetes era muy repetitivos, lo primero que debíamos hacer era ingresar al directorio /mnt/lfs/sources con el comando `cd $LFS/sources`, luego listábamos (`ls`) los paquetes (comprimidos) descargados anteriormente en el capítulo 3(*wget-list*), el siguiente paso era descomprimir con el comando `tar -xvf <Nombre del paquete comprimido>`, una vez descomprimido ingresábamos al directorio del paquete y luego seguíamos las instrucciones del libro para la configuración, compilación e instalación. Una vez terminadas las instrucciones debíamos borrar el paquete descomprimido (si el libro no indicaba lo contrario).

Los paquetes instalados en el capítulo 5 fueron:

- **Binutils-2.37 – Paso 1**
- **GCC-11.2.0 - Paso 1**
- **Encabezados de API Linux-5.13.10**
- **Glibc-2.34**
- **Libstdc ++ de GCC-11.2.0, paso 1**

Una vez llegado al paquete Libstdc ++ de GCC-11.2.0 nos encontramos con un error, el cual se debía a una mala compilación del paquete GCC, por cual debimos volver a hacer la instalación correspondiente del paquete, luego pudimos avanzar sin errores.

Ingresando al capítulo 6, seguimos con la instalación de los paquetes, con los mismos pasos citados anteriormente, los paquetes instalados con fueron:

- **M4-1.4.19**
- **Ncurses-6.2**
- **Bash-5.1.8**
- **Coreutils-8.32**
- **Diffutils-3.8**
- **Archivo-5.40**
- **Findutils-4.8.0**
- **Gawk-5.1.0**
- **Grep-3.7**

- **Gzip-1.10**
- **Hacer-4.3**
- **Parche-2.7.6**
- **Sed-4.8**
- **Alquitrán-1.34**
- **Xz-5.2.5**
- **Binutils-2.37 – Paso 2**
- **GCC-11.2.0 - Paso 2**

En el capítulo 7 empezamos a trabajar dentro del entorno “chroot”, en el cual trabajamos completamente aislado del sistema operativo host, a excepción del kernel en ejecución. Para ello, debimos cambiar la propiedad de los directorios, pasar del usuario lfs (un usuario que existe solo en el sistema host) al usuario root.

```
chown -R root:root $LFS/{usr,lib,var,etc,bin,sbin,tools}
case $(uname -m) in
x86_64) chown -R root:root $LFS/lib64 ;;
esac
```

Así también debimos crear otros directorios en los cuales más adelante se montarían los sistemas de archivos.

```
mkdir -pv $LFS/{dev,proc,sys,run}
```

Luego creamos nodos de dispositivos iniciales ya que cuando el núcleo arranca el sistema, requiere la presencia de algunos nodos de dispositivos, también montamos un sistema de archivos de kernel virtual ya que este es el método recomendado para llenar el /dev directorio con dispositivos.

Cabe resaltar que estos sistemas de archivos deben ser montados cada vez que se abandona el entorno “chroot”

```
mount -v --bind /dev $LFS/dev
mount -v --bind /dev/pts $LFS/dev/pts
mount -vt proc proc $LFS/proc
mount -vt sysfs sysfs $LFS/sys
mount -vt tmpfs tmpfs $LFS/run
```

Para ingresar al entorno “chroot”, se debe utilizar el siguiente comando

```
chroot "$LFS" /usr/bin/env -i \  
  HOME=/root \  
  TERM="$TERM" \  
  PS1='(lfs chroot) \u:\w\$ ' \  
  PATH=/usr/bin:/usr/sbin \  
  /bin/bash --login +h
```

A continuación, seguimos todos los pasos mencionados en el libro para la configuración del entorno. Así también procedimos a la instalación de los paquetes sin inconveniente alguno.

Los paquetes instalados en el capítulo 7 fueron:

- **Libstdc++ from GCC-11.2.0, Paso 2**
- **Gettext-0.21**
- **Bison-3.7.6**
- **Perl-5.34.0**
- **Python-3.9.6**
- **Texinfo-6.8**
- **Util-linux-2.37.1**

Por último, al final el capítulo , se realiza un backup ya que en el Capítulo 8 , los archivos temporales se sobrescriben.

```
cd $LFS  
tar -cJpf $HOME/lfs-temp-tools-11.0-rc1.tar.xz .
```

Construyendo el Sistema LFS

Tercera y última etapa. Comenzamos con el capítulo 8, en el cual seguimos con la instalación y configuración de los distintos paquetes, con los pasos citados anteriormente en el capítulo 5. Entre estos paquetes se encuentra “Systemd” el contiene programas para controlar el inicio, la ejecución y el apagado del sistema. Este paquete lo compilamos dos veces ya que en la primera vez presento errores. Una vez compilado por segunda vez, los errores ya no fueron percibidos. Los demás paquetes no presentaron errores.

Los paquetes instalados en este capítulo fueron:

- **Man-pages-5.12**
- **lana-Etc-20210611**
- **Glibc-2.34**
- **Zlib-1.2.11**
- **Bzip2-1.0.8**
- **Xz-5.2.5**
- **Zstd-1.5.0**
- **File-5.40**
- **Readline-8.1**
- **M4-1.4.19**
- **Bc-5.0.0**
- **Flex-2.6.4**
- **Tcl-8.6.11**
- **Expect-5.45.4**
- **DejaGNU-1.6.3**
- **Binutils-2.37**
- **GMP-6.2.1**
- **MPFR-4.1.0**
- **MPC-1.2.1**
- **Attr-2.5.1**
- **Acl-2.3.1**
- **Libcap-2.52**
- **Shadow-4.9**
- **GCC-11.2.0**
- **Pkg-config-0.29.2**
- **Ncurses-6.2**
- **Sed-4.8**
- **Psmisc-23.4**
- **Gettext-0.21**
- **Bison-3.7.6**
- **Grep-3.7**
- **Bash-5.1.8**
- **Libtool-2.4.6**
- **GDBM-1.20**
- **Gperf-3.1**
- **Expat-2.4.1**
- **Inetutils-2.1**
- **Less-590**
- **Perl-5.34.0**
- **XML::Parser-2.46**

-
- Intltool-0.51.0
 - Autoconf-2.71
 - Automake-1.16.4
 - Kmod-29
 - Libelf from Elfutils-0.185
 - Libffi-3.4.2
 - OpenSSL-1.1.1k
 - Python-3.9.6
 - Ninja-1.10.2
 - Meson-0.59.0
 - Coreutils-8.32
 - Check-0.15.2
 - Diffutils-3.8
 - Gawk-5.1.0
 - Findutils-4.8.0
 - Groff-1.22.4
 - GRUB-2.06
 - Gzip-1.10
 - IPRoute2-5.13.0
 - Kbd-2.4.0
 - Libpipeline-1.5.3
 - Make-4.3
 - Patch-2.7.6
 - Tar-1.34
 - Texinfo-6.8
 - Vim-8.2.3337
 - MarkupSafe-2.0.1
 - Jinja2-3.0.1
 - Systemd-249
 - D-Bus-1.12.20
 - Man-DB-2.9.4
 - Procps-ng-3.3.17
 - Util-linux-2.37.1
 - E2fsprogs

Al terminar el capítulo 8, se actualiza el comando para ingresar al entorno “chroot” ya que aquí la +h (imprime la información de ubicación de los comandos encontrados.) ya no se usa, ya que todos los programas anteriores han sido reemplazados.

El capítulo 9 trata sobre los archivos de configuración y los servicios systemd. En primer lugar, se presentan los archivos de configuración general necesarios para configurar la red. Aquí definimos el nombre del host "araujo.olmedo". Luego se analizan los problemas que afectan la configuración adecuada de los dispositivos. También se muestra la configuración del reloj del sistema y la disposición del teclado. En tercer lugar, se presentan los scripts y archivos de configuración utilizados cuando el usuario inicia sesión en el sistema. Por último, se ve la configuración del comportamiento de systemd.

En el capítulo 10 es hora de hacer que el sistema LFS sea de arranque. Este capítulo trata sobre la creación del /etc/fstab archivo

```
cat > /etc/fstab << "EOF"
# Begin /etc/fstab
# file system  mount-point  type    options          dump  fsck
#                                     order
/dev/sdb1    /                ext4    defaults         1    1
/dev/sdb2    swap            swap    pri=1            0    0
# End /etc/fstab
EOF
```

La construcción de un núcleo para el nuevo sistema LFS, descomprimiendo el paquete Linux y la instalación del cargador de arranque GRUB, el cual instalamos en el sdb1 para que el sistema LFS pueda seleccionarse para arrancar al inicio.

En el capítulo 11 desmontamos el sistema de archivos LFS y reiniciamos el sistema.

A pesar de todos realizar nuestro mayor esfuerzo, desafortunadamente no pudimos hacer que el LFS arranque independientemente del anfitrión. Volvimos a realizar el LFS varias veces, en varias máquinas diferentes y el resultado siempre fue el mismo. Un mensaje de Kernel Panic – not syncing: No working init found

```
[ 1.560692] Freeing unused kernel image (rodata/data gap) memory: 1144K
[ 1.561018] Run /sbin/init as init process
[ 1.564179] Run /etc/init as init process
[ 1.565428] Run /bin/init as init process
[ 1.566667] Run /bin/sh as init process
[ 1.568128] Kernel panic - not syncing: No working init found. Try passing i
init= option to kernel. See Linux Documentation/admin-guide/init.rst for guidance
[ 1.569929] CPU: 0 PID: 1 Comm: swapper/0 Not tainted 5.13.10 #1
[ 1.570493] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBo
x 12/01/2006
[ 1.570658] Call Trace:
[ 1.570664] dump_stack+0x64/0x7c
[ 1.571358] ? rest_init+0x80/0xa4
[ 1.571896] panic+0xf6/0x2b7
[ 1.572400] ? kernel_execve+0x13c/0x1a0
[ 1.572917] ? rest_init+0xa4/0xa4
[ 1.573376] kernel_init+0xf7/0xfc
[ 1.573817] ret_from_fork+0x22/0x30
[ 1.574380] Kernel Offset: 0x16600000 from 0xffffffff81000000 (relocation ran
ge: 0xffffffff80000000-0xffffffffbfffffff)
[ 1.575317] ---[ end Kernel panic - not syncing: No working init found. Try
passing init= option to kernel. See Linux Documentation/admin-guide/init.rst for
guidance. ]---
```

Estuvimos trabajando muy de cerca con los ayudantes de catedra (José y Cristóbal), intentando varias soluciones sin éxito alguno.

Algunas de las soluciones que intentamos fueron volver a compilar el paquete Linux, donde se configuraba el kernel, volver a compilar el paquete Systemd, cambiar la configuración del grub.cfg, donde se encontraba el menuentry. Volver a instalar un nuevo host y montar allí el LFS, comparamos nuestros archivos con otros grupos y no encontramos diferencia alguna. Buscamos el archivo init, y se encontraba en el lugar correspondiente /etc/init.d. También ampliamos el tamaño de la memoria RAM (recomendación de un foro), cambiamos el orden de arranque.

Como podrá leer en la documentación, no tuvimos mayores inconvenientes al momento de realizar las instalaciones y configuraciones del sistema, por lo cual no sabemos el motivo del porque no realiza correctamente el boot.