Universidad del Valle de Guatemala
Sistemas Operativos
Catedrático: Tomás Gálvez
Semestre 1, 2020

Proyecto de semestre: LFS

Fecha de Entrega: última semana de clases.

Descripción: este proyecto reta al estudiante a armar (*put together*) su propio sistema operativo Linux siguiendo la guía del proyecto *Linux From Scratch,* versión 9.0 (http://www.linuxfromscratch.org/lfs/). Se plantean preguntas en diferentes etapas del desarrollo, con cuya respuesta el/la estudiante profundizará sus conocimientos sobre los componentes esenciales y opcionales de un sistema operativo, así como su relación y experiencia con Linux. Al final del semestre, el/la estudiante deberá presentar su LFS funcionando en una máquina virtual o computadora física, así como un documento bien organizado que presente respuestas a las preguntas planteadas en este proyecto.

Materiales: necesitará un sabor de Linux donde desarrollar su LFS. Se recomienda un sabor popular como Ubuntu, Arch, Debian, etc., por presentar mayor compatibilidad y más fuentes de documentación.

Nota: iteraciones previas del curso han desarrollado el LFS sobre el sistema operativo Knoppix, versión 7.2 de 32 bits. Las ventajas han sido que ha funcionado y no es muy grande, por lo que se descarga rápido y ocupa poco espacio. Sin embargo, es una versión antigua de un sistema operativo con escasa documentación en línea. Además, los repositorios para descarga de las herramientas al inicio del proyecto están desactualizados. Actualizarlos no es necesariamente complicado, pero hay que buscar y probar.

IMPORTANTE: sin importar el sistema operativo que use como base, se recomienda trabajar el proyecto en una máquina virtual y crear muchos *snapshots*. El proyecto le exigirá realizar operaciones que pueden poner en riesgo el correcto funcionamiento de su computadora. Cada estudiante será únic@ responsable de lo que suceda con el equipo sobre el que trabaje.

Nota: a lo largo de este documento se plantean preguntas relacionadas con las diferentes etapas de trabajo durante el desarrollo del LFS. Las preguntas están organizadas por secciones con el fin de proveer puntos de referencia al procedimiento descrito en los capítulos de la guía oficial de LFS. Las secciones como tales no tienen ningún otro propósito.

Nota: necesitará copiar muchos comandos a lo largo de la creación del LFS. No se recomienda copiar comandos del libro en PDF porque provoca errores. Se recomienda copiar los comandos a mano y almacenarlos en un archivo para ejecución controlada. Alternativamente, se pueden copiar de la versión HTML del libro.

Universidad del Valle de Guatemala
Sistemas Operativos
Catedrático: Tomás Gálvez
Semestre 1, 2020

Preguntas por sección

Universidad del Valle de Guatemala
Sistemas Operativos
Catedrático: Tomás Gálvez
Semestre 1, 2020

Listado de preguntas

Universidad del Valle de Guatemala
Sistemas Operativos
Catedrático: Tomás Gálvez
Semestre 1, 2020

Instalación de herramientas necesarias para la creación de un LFS

Durante la instalación inicial de herramientas deberá ejecutar un *script* llamado `version-check.sh`. Revise el código de este *script* para responder lo siguiente:

- ¿Por qué las condiciones de los *if*'s están entre corchetes?
- ¿Cuáles son las condiciones que se están verificando en cada caso?
- ¿Para qué sirve '`$`' en *shell code*?

A lo largo del proyecto se instalarán muchas herramientas siguiendo la manera estándar en Linux (es decir, sin uso de un manejador de paquetes). Estas instrucciones suelen incluirse con los paquetes en un archivo llamado INSTALL. Descargue uno de los paquetes listados como requerimientos para el LFS (es probable que necesite descargar uno o más para actualizarlos, de cualquier manera) y vea el contenido de INSTALL.

- ¿Cuáles son los tres comandos que se deben ejecutar para instalar un paquete en Linux?
- ¿Dónde encontramos información adicional de instalación específica para cada paquete?

Parte de este proceso de instalación involucrará descargar y extraer las *tarballs* de las herramientas.

- ¿Qué es una *tarball*?
- ¿Por qué las *tarball* tienen una extensión "doble" (*e.g.,* `.tar.gz`)?

La extracción de una *tarball* suele hacerse con el comando `tar -xvzf` seguido del archivo que se desea descomprimir.

- ¿Qué significa cada una de las opciones (`-xvzf`)?
- ¿Cuál de estas opciones podría haberse omitido? ¿Cuál de estas opciones cambia si la extensión del archivo comprimido es diferente a `.tar.gz`?

Otra operación que se realizará repetidas veces durante el proyecto es la creación o manejo de enlaces simbólicos.

- ¿Qué es un enlace simbólico?
- ¿Cuál es la diferencia entre un enlace simbólico (*symlink*) y uno duro (*hard link*)?
- En el comando `ln` usado para crear enlaces, ¿en qué orden se deben escribir los parámetros para crear un enlace de *A* a *B*?

Universidad del Valle de Guatemala
Sistemas Operativos
Catedrático: Tomás Gálvez
Semestre 1, 2020

Preparación de un ambiente aislado para creación de un LFS

En el capítulo 4 de la guía de desarrollo de un LFS inicial la creación de un ambiente aislado dentro del sistema operativo *host* (es decir, en el cual se armará el LFS) para compilar e instalar las herramientas del LFS sin ligarlas al sistema en el que se compilan. Parte de esto involucra crear un archivo llamado `.bash_profile` para declarar variables de ambiente y montar las particiones donde se construirá el LFS.

- ¿Qué hacen los archivos `.bash_profile`?
- ¿Para qué sirve crear un archivo llamado `.profile`?

La creación del ambiente aislado involucra también crear un usuario llamado `lfs` con el siguiente comando:

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

- ¿Qué es `/dev/null`?
- Explique las opciones y parámetros usados en este comando.

Más adelante se le otorga a este usuario la propiedad de algunos directorios con el comando `chown`.

- ¿Qué significa ser dueño de un directorio?

Luego de estos cambios de propiedad deberá iniciar una sesión usando el nombre `lfs`.

- ¿Cuál es la diferencia entre una *login shell* y una *non-login shell*?

Creará un archivo que permita ejecutar una *shell* por defecto para el usuario `lfs`. Esta *shell* proveerá un nuevo ambiente en el que sólo estén definidas tres variables: el directorio `HOME`, el tipo de terminal que se usa y lo que se escribe en el *prompt* de la terminal (lo que aparece siempre antes del cursor). Para hacerlo creará un archivo llamado `.bash_profile` que contendrá el siguiente comando:

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

- Explique exactamente qué se está haciendo con `env -i HOME=$HOME TERM=$TERM PS1='\u:\w\$ ' /bin/bash`.
- ¿Cuál es el efecto o propósito de preceder el comando anterior con `exec`?

Por último, creará un archivo llamado `.bashrc` conteniendo las líneas `set +h` y `umask 022`. `set +h` apaga la funcionalidad de *hashing* para la *shell*.

- ¿Para qué sirve el *hashing* de la *shell*?
- ¿Qué hace el comando `umask` y cuál es el efecto de usarlo con el parámetro `022`?

Universidad del Valle de Guatemala
Sistemas Operativos
Catedrático: Tomás Gálvez
Semestre 1, 2020

Instalación de *toolchain* para creación de un LFS

Primera instalación de Binutils

Luego de la explicación sobre el propósito de la *toolchain*[1] y las importantes instrucciones de compilación, el capítulo 5 inicia una serie de instalaciones. Se configurará el paquete Binutils para instalación especificando, entre otras, las siguientes opciones y valores correspondientes: `--target=$LFS_TGT` y `--with-sysroot=$LFS`. Ambas de las variables de ambiente usadas como valores debieron ser definidas en pasos anteriores.

- ¿Qué se logra (más allá de los efectos en la configuración) al especificar `--with-sysroot=$LFS`?
- ¿Por qué se especifica `$LFS_TGT` como el valor de `--target`?
- ¿Qué es la *target triplet* y cuál es su campo *vendor*?
- ¿Cuál es el significado o proósito del campo *vendor,* y para qué sirve le asignamos el valor `lfs`?
- ¿Cómo se relaciona la *target triplet* con la opción `--prefix=/tools`, también usada?

Primera instalación de GCC

Antes de configurar GCC para instalación ejecutará el siguiente comando:

```
for file in gcc/config/{linux,i386/linux{,64}}.h
do
  cp -uv $file{,.orig}
  sed -e 's@/lib\(64\)\?\(32\)\?/ld@/tools&@g' \
      -e 's@/usr@/tools@g' $file.orig > $file
  echo '
#undef STANDARD_STARTFILE_PREFIX_1
#undef STANDARD_STARTFILE_PREFIX_2
#define STANDARD_STARTFILE_PREFIX_1 "/tools/lib/"
#define STANDARD_STARTFILE_PREFIX_2 ""' >> $file
  touch $file.orig
done
```

- ¿Para qué sirve encerrar una lista de *strings* entre llaves, en este caso?
- ¿Qué hace el programa `sed`? Aclare con el mayor detalle posible qué está haciendo `sed` con ese relajo abajo de `cp -uv $file{,.orig}`.
- ¿Qué hace la instrucción `touch`?
- ¿Qué hace la opción `-u` de `cp` y cómo se relaciona con `touch` en este ciclo?
- ¿Para qué sirven las macros `STANDARD_STARTFILE_PREFIX_N` y por qué se les asignan los valores que se les asignan en este ciclo?
- Desglose y explique el funcionamiento de este ciclo.

En la instrucción de configuración para GCC:

[1]: esta explicación suele ser enredada y confusa la primera vez que se lee, pero es recomendado leerla de igual forma. La sección 5.2 es más útil como referencia a lo largo del capítulo 5.

- ¿Qué hace la opción `--disable-nls`?
- ¿Qué hace la opción `--disable-shared` y por qué es necesaria? ¿Cuál es la diferencia entre *linking* estático y dinámico?

Instalación de Glibc

Durante la configuración para instalación de Glibc se especifica la opción `build` con el resultado de ejecutar un *script* llamado `config.guess` como valor asignado.

- ¿Qué hace la ejecución del *script* `config.guess` en `glibc-2.30/scripts`?
- ¿Cuál es el efecto de las opciones `--host` y `--build` en la configuración?

Sanity check

Es importante realizar las pruebas de funcionalidad básica de los paquetes recién instalados, antes de continuar. Entre las instrucciones para ello se usa la herramienta de Binutils `readelf`.

- ¿Qué es un archivo ELF y qué hace la opción `-l` de `readelf`?

Hasta el momento se han instalado GCC y Binutils con configuraciones que conforman lo que el libro llama *cross GCC* y *cross Binutils*. Luego de esto instalará el paquete Libstdc++.

- ¿Qué es un *cross compiler* y por qué lo necesitamos? Se recomienda leer la sección 5.2 del libro de LFS.
- ¿Cuál es el efecto de compilar con `$LFS_TGT-gcc` en lugar de sólo `gcc`?
- Explique la relación entre Binutils, GCC, Glibc y Libstdc++.
- ¿Por qué se usa la opción `--with-gxx-include-dir=/tools/$LFS_TGT/include/c++/9.2.0` en la configuración?

Segunda instalación de Binutils

- ¿Por qué se debe realizar una segunda instalación de Binutils?
- ¿Qué hacen las herramientas `ranlib` y `ar`, de Binutils?

Segunda instalación de GCC

Creará un archivo llamado `limits.h` en un directorio armado con un comando de *shell scripting*: `dirname`. Este comando se ejecutará en una *subshell* mediante substitución de comandos con ‘`’.

- ¿Para qué sirve el comando `dirname`?
- ¿Cuál es la diferencia entre ejecutar algo con `$` y ejecutarlo con `` ` ``?
- ¿Qué contiene el *header* `limits.h`? Normalmente GCC ejecuta este mismo comando durante su instalación y el resultado es un *header* llamado `limits.h` que incluye otro *header* llamado `limits.h` provisto por el sistema. ¿Por qué la primera instalación de GCC no hizo esto desde un principio?

El comando de configuración de GCC que se usará empleará la opción `--disable-bootstrap` que deshabilita el comportamiento habitual que tiene GCC de compilarse haciendo un *bootstrap build*.

- ¿Qué es un *bootstrap build*?

Instalará un paquete llamado Tcl , otro llamado Expect y otro más llamado DejaGNU.

- ¿Qué es Tcl y con qué propósito fue creado?
- ¿Para qué sirve Expect?
- ¿Cuál es la relación entre DejaGNU, Expect y Tcl?

Instalará otros paquetes llamados: M4, Ncurses, Bash, Coreutils, Diffutils, Gettext, Make y Patch. En las opciones de configuración de Bash se especificará `--without-bash-malloc`. Durante la instalación del paquete Gettext será necesario que compile los programas `msgfmt`, `msgmerge` y `xgettext`. Y para la configuración de Make usará la opción `--without-guile`.

- ¿Para qué sirve Ncurses?
- ¿Qué es `malloc`?
- ¿Para qué sirve M4?
- ¿Qué hace el programa `hostname` de Coreutils?
- ¿Cuál es la relación entre Patch y Diffutils?
- ¿Cuál es la relación entre los programas `msgfmt`, `msgmerge` y `xgettext` de Gettext?
- ¿Qué es GNU Guile?
- Investigue y explique qué son las *automatic variables* en un *makefile*.

Al completar las instalaciones para construcción del Linux temporal procederá con la remoción de *debugging symbols* usando los siguientes comandos:

```
strip --strip-debug /tools/lib/*
/usr/bin/strip --strip-unneeded /tools/{,s}bin/*
rm -rf /tools/{,share}/{info,man,doc}
```

- ¿Qué son *debugging symbols*?
- ¿Para qué sirve `strip`?
- ¿Por qué el primer comando simplemente ejecuta `strip` pero el segundo ejecuta `/usr/bin/strip`?
- Explique el concepto de *relocation* asociado a los *linkers*, y su relación con la opción `--strip-unneeded`.

Universidad del Valle de Guatemala
Sistemas Operativos
Catedrático: Tomás Gálvez
Semestre 1, 2020

Construcción de LFS (parte I)

En el capítulo 6, los primeros pasos de construcción de un LFS involucrarán montar sistemas de archivos en directorios de nuestro ambiente temporal.

- ¿Qué es un sistema de archivos? ¿Qué es un sistema de archivos virtual?
- ¿Qué significa "montar" un sistema de archivos?
- Explique el propósito de cada uno de los sistemas de archivos `/dev`, `/proc`, `/sysfs` y `/run`.
- Explique a detalle qué hacen los comandos siguientes:

```
sudo mknod -m 600 $LFS/dev/console c 5 1
sudo mknod -m 666 $LFS/dev/null c 1 3
```

- Explique a detalle qué hacen los comandos `mount` siguientes:

```
sudo mount -vt devpts devpts $LFS/dev/pts -o gid=5,mode=620
sudo mount -vt proc proc $LFS/proc
sudo mount -vt sysfs sysfs $LFS/sys
sudo mount -vt tmpfs tmpfs $LFS/run
```

Eventualmente accederá a un ambiente temporal para iniciar la construcción del LFS. Para ello se usará un comando como el siguiente:

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

- Explique exactamente qué hace el comando anterior y por qué es importante para el resto del proyecto.

Sanity check

En este punto es importante realizar un paso de revisión que el libro de LFS no incluye, pero cuyo potencial problema nos impedirá continuar más adelante. Cree un programa de prueba en C con el siguiente comando:

```
echo 'int main(){}' > dummy.c
```

Ahora compílelo usando el comando `cc`:

```
cc dummy.c
```

Esto no debería producir ningún error. A continuación, diríjase al directorio `/tools/i686-pc-linux-gnu/bin` y ejecute el siguiente comando:

```
readelf -e as | grep interpreter
```

El resultado debería ser el siguiente:

```
[Requesting program interpreter: /tools/lib/ld-linux.so.2]
```

Si el resultado no incluye `/tools`, el ensamblador instalado previamente no fue correctamente configurado. Este es un paso crucial realizado temprano en la creación de la *toolchain*, por lo que será necesario repetir toda su construcción (capítulo 5 del libro de LFS) antes de continuar (específicamente, hay que repetir la primera instalación de Binutils, pero esa instalación es la primera que hacemos en la creación de la *toolchain* y todo lo que instalamos después depende de ella).

Si todo está bien, proceda a eliminar `dummy.c` y `a.out` del directorio donde los creó. **Nota importante**: como se indica claramente en el libro de LFS, si salimos del ambiente `chroot` creado será necesario volver a entrar a él antes de continuar con cualquier paso a partir de este punto. Si reiniciamos la máquina virtual (o de alguna otra forma desmontamos los sistemas de archivos virtuales), también deberemos volver a montar los sistemas de archivos virtuales antes de ejecutar el `chroot`.

Como siguientes pasos se crearán varios directorios y algunos *links* simbólicos. En particular, se crearán *links* simbólicos apuntando hacia directorios contenidos en `/tools`. Algunos de los directorios se crearán con el comando `install`.

- ¿Por qué es necesario crear un conjunto de *links* simbólicos que apuntan de directorios como `/bin`, `/usr/bin` y `/usr/lib` a `/tools/bin` y `/tools/lib`, correspondientemente?
- ¿Para qué sirve el comando `install`?
- ¿Para qué sirve el archivo `/etc/mtab` y por qué se *linkea* a `/proc/self/mounts`?
- ¿Qué es el *Filesystem Hierarchy Standard* (FHS)?

Para construir el LFS se crearán usarios y grupos de usuarios en este sistema, lo que involucra crear dos archivos: `/etc/passwd` y `/etc/group`. Luego de crearse el archivo `/etc/passwd` se ejecutará el siguiente comando:

```
exec /tools/bin/bash --login +h
```

- ¿Para qué sirven los archivos `/etc/passwd` y `/etc/group`? Explique la sintaxis del contenido de estos archivos.
- Explique el propósito y configuración de los usuarios creados en `/etc/passwd`.
- ¿Cuál es el propósito de ejecutar el comando `exec` mostrado arriba, y por qué se logra ese propósito usando `exec` en lugar de sólo `/tools/bin/bash`?

Luego de unos pasos más para la configuración de bitácoras instalaremos varias herramientas en nuestro ambiente aislado. Lo primero es instalar los API *headers* de Linux y luego el paquete de *man pages*. Posterior a esto…

Universidad del Valle de Guatemala
Sistemas Operativos
Catedrático: Tomás Gálvez
Semestre 1, 2020

Instalación de Glibc

Se comienza por aplicar un parche al paquete con el siguiente comando:

```
patch -Np1 -i ../glibc-2.30-fhs-1.patch
```

- Dé un vistazo al contenido del archivo `glibc-2.30-fhs-1.patch` y responda: ¿qué hace el programa `patch`?
- ¿Para qué sirven las opciones `-Np1` e `-i` especificadas?

Luego de ejecutarse la *test suite* de Glibc se requerirá la ejecución de un comando `touch` y un comando `sed` antes de proceder a instalar. Luego de la instalación de Glibc se instalarán también un archivo de configuración y un directorio de ejecución para el programa `nscd`.

- ¿Qué hace `touch` cuando recibe un archivo inexistente como argumento?
- ¿Qué es `nscd`?

Posterior a la configuración de `nscd` se instalarán `locales`, configuraciones de idioma y moneda para el sistema. Luego se crea un archivo de configuración para `nsswitch`.

- ¿Cuál es el propósito de `nsswitch.conf`?
- ¿Cuál es el propósito de cada una de las bases de datos configuradas en este archivo?

Durante la instalación de la *data* de horario y ubicación se empleará el comando `zic` y se manipularán los archivos `zone.tab`, `zone1970.tab` e `iso3166.tab`.

- ¿Para qué sirve el comando `zic`?
- ¿Cuál es el propósito de los archivos `zone.tab`, `zone1970.tab` e `iso3166.tab`?

Como últimos pasos se alterarán algunos directorios y se modificarán las especificaciones de GCC para que se adapten al uso de este nuevo *loader,* con el siguiente comando:

```
gcc -dumpspecs | sed -e 's@/tools@@g' \
-e '/\*startfile_prefix_spec:/{n;s@.*@/usr/lib/ @}' \
-e '/\*cpp:/{n;s@$@ -isystem /usr/include@}' > \
`dirname $(gcc --print-libgcc-file-name)`/specs
```

- ¿Qué hace la instrucción `n` en `sed`?
- ¿Cuál es el propósito del archivo `specs`?

Sanity check

Se realizará un nuevo *sanity check* para garantizar que la instalación de las herramientas hasta ahora se realizó correctamente. Esto es muy importante para garantizar la correcta compilación de las próximas instalaciones. Se compilará un archivo de prueba con las siguientes instrucciones:

```
cc dummy.c -v -Wl,--verbose &> dummy.log
```

- ¿Qué hace la opción `-Wl` en el comando `cc`?
- ¿Para qué sirve usar `&>`?

Los resultados de la compilación serán analizados en el archivo `dummy.log`. Para el análisis se buscarán palabras clave con el programa `grep`. El primer comando es el siguiente:

```
grep -o '/usr/lib.*/crt[1in].*succeeded' dummy.log
```

El resultado debería ser el siguiente:

```
/usr/lib/crt1.o succeeded
/usr/lib/crti.o succeeded
/usr/lib/crtn.o succeeded
```

Luego se ejecuta:

```
grep -B1 '^ /usr/include' dummy.log
```

Y el resultado debería ser:

```
#include <...> search starts here:
/usr/include
```

A continuación, las demás instrucciones del *sanity check* y sus correspondientes resultados esperados:

```
grep 'SEARCH.*/usr/lib' dummy.log |sed 's|; |\n|g'
```

```
...
SEARCH_DIR("/usr/lib")
SEARCH_DIR("/lib");
...
```

```
grep "/lib.*/libc.so.6 " dummy.log
```

```
attempt to open /lib/libc.so.6 succeeded
```

```
grep found dummy.log
```

```
found ld-linux.so.2 at /lib/ld-linux.so.2
```

- ¿Cuál es el efecto de los símbolos `.`, `[]` y `*` en los argumentos de `grep`?
- ¿Qué hace la opción `-o` de `grep`?
- Explique el propósito de los archivos `crt1.o`, `crti.o` y `crtn.o`.
- En el comando `grep` con la opción `-B1`, ¿qué sucedería si no usamos esa opción?

- De acuerdo con estos resultados, ¿dónde se inicia la búsqueda de encabezados que #incluimos en nuestros programas?
- ¿Qué es y para qué sirve un *soname*? Explique qué son `libc.so.6` y `ld-linux.so.2` (se recomienda investigar el concepto de *shared library*).
- ¿Qué es *underlinking* y por qué se considera mala práctica?

Instalación de Binutils

Durante la instalación de Binutils se verificará el funcionamiento de las PTY's con la siguiente instrucción:

```
expect -c "spawn ls"
```

Debería obtenerse el siguiente resultado:

```
spawn ls
```

Si se obtiene el siguiente texo, hay un problema:

```
The system has no more ptys.
Ask your system administrator to create more.
```

En este caso deberá regresar y repetir los pasos que sean necesarios para evitarlo.

- Investigue:
  - ¿En qué consiste una PTY?
  - ¿Cómo funcionan las PTY?
  - ¿Cómo se relacionan con `expect`?
  - ¿Cómo se relacionan con el sistema de archivos `devpts` que montamos casi al inicio de la construcción del LFS?
  - ¿Cómo se relacionan con los emuladores de terminales (investigue este concepto)?

Se ejecutará la *test suite* usando la opción `-k` del comando `make`.

- ¿Qué hace la opción `-k`?

Luego de Binutils se instalará GMP y se ejecutará su *test suite* con el siguiente comando:

```
make check 2>&1 | tee gmp-check-log
```

- ¿Qué diferencia hay entre usar `&>` y `2>&1`, y cuál es el efecto de esto último combinado con el comando `tee`?

Antes de la nueva instalación de GCC se instala el paquete Shadow para manejo seguro de contraseñas.:

- ¿Qué son `/var/spool/mail` y `/var/mail`; y cuál es su relación con Shadow?

Universidad del Valle de Guatemala
Sistemas Operativos
Catedrático: Tomás Gálvez
Semestre 1, 2020

Instalación de GCC

Luego de compilarse GCC se ejecutará la *test suite*. Tome en cuenta que esto toma mucho tiempo, así que procure dejar la máquina virtual corriendo durante toda la noche. No olvide el nuevo *sanity check*.

Si no ha hecho *snapshots,* este es un buen momento para hacer uno. Cuando la *test suite* termine su trabajo y luego de realizar la instalación del paquete se creará un *link* simbólico para permitir el uso de *Link Time Optimization* (LTO).

- ¿Qué es y para qué sirve LTO?

Eventualmente se instalará el paquete Pkg-config.

- ¿Para qué sirve Pkg-config?

Luego de Pkg-config se instalará Ncurses. Luego de instalarse este paquete se ejecutarán las siguientes instrucciones para asegurar que aplicaciones que funcionan sin *wide-character libraries* (que se instalaron al configurar este paquete incluyendo la opción `--enable-widec`) usen las *libraries* instaladas:

```
for lib in ncurses form panel menu ; do
      rm -vf /usr/lib/lib${lib}.so
      echo "INPUT(-l${lib}w)" > /usr/lib/lib${lib}.so
      ln -sfv ${lib}w.pc /usr/lib/pkgconfig/${lib}.pc
done
```

- Lea el contenido de uno de los archivos creados con la instrucción anterior (*e.g.* usando `cat /usr/lib/libncurses.so`). ¿Qué hace la instrucción allí contenida?
- ¿Qué es un *linker script* y para qué sirve?

Se ejecutarán otras instrucciones con el similar propósito de obligar a que las aplicaciones que requieren Curses usen el paquete recién instalado

- ¿Cuál es la diferencia entre Curses y Ncurses?

Después de Ncurses se instala el paquete Attr.

- ¿Para qué sirve este paquete? Incluya una breve explicación de qué son los *extended attributes* en un sistema de archivos.

La instalación de Attr es seguida de la instalación de Acl para manejo de listas de control de acceso, y luego por la instalación de Libcap para manejo de *capabilities*.

- ¿Qué son las listas de control de acceso (ACL's)?
- Explique la relación entre listas de control de acceso y *capabilities*.

Se instalarán varios paquetes luego de Libcap, incluyendo Flex y Libtool.

- ¿Cuál es la diferencia entre Lex y Flex?
- ¿Cuál es el propósito/beneficio que provee el *GNU generic library support script*?

Universidad del Valle de Guatemala
Sistemas Operativos
Catedrático: Tomás Gálvez
Semestre 1, 2020

Construcción de LFS (parte II)

Durante la instalación del paquete Inetutils en la sección 6.39, será necesario mover algunos programas a los directorios `/bin` y `/sbin`.

- ¿Cuál es el propósito del directorio `/sbin`?

Instalación de Perl

Para configurar la instalación Perl es necesario crear el archivo `/etc/hosts`. La configuración en sí se realiza mediante un comando diferente a los usados para instalaciones anteriores. El *script* de configuración recibe las opciones `-des` y las opciones de configuración comienzan con `D`. Una de las opciones sirve para configurar el *pager* de Perl.

- ¿Cuál es el propósito del archivo `/etc/hosts`?
- ¿Qué hacen las opciones `-des` y qué diferencia habría si no se usan?
- El libro indica que la opción `-Dpager` de arriba sirve para usar `less` en lugar de `more`. ¿Qué es el *pager* de Perl?

Instalación de Autoconf y Automake

- Describa el propósito de, y la relación entre, Autoconf y Automake.

Luego de Automake se instalarán varias otras herramientas, incluyendo Gawk, Groff y Texinfo.

- Gawk es la implementación GNU de Awk. Describa Awk.
- ¿Cuál es la importancia de Groff (y sus predecesores) en el contexto de los sistemas Unix y sus derivados?
- Explique brevemente la diferencia entre *man pages* e *info pages*.

Todos hemos usado (o usaremos) el editor de texto Vi. La instalación de este programa, mediante el paquete Vim, requiere especificar la ubicación por defecto del archivo `vimrc`.

- ¿Qué es `vimrc`?
- ¿Qué ventajas presenta Vim sobre Vi?

Después de Vim toca Procps-ng. Para este paquete se hacen unas modificaciones que evitan fallos de pruebas que requieren el uso de dispositivos `tty`.

- ¿Qué es un dispositivo `tty`?

Un importante paquete a instalarse es Util-linux. La instalación comienza por crear algunos directorios para almacenar `adjtime` y `hwclock`.

- ¿Qué hacen el archivo de configuración `adjtime` y el programa `hwclock`?

Al instalarse el paquete Util-linux se hará sin soporte para `systemd`, el programa que hemos hablado en clase que reemplaza al tradicional `init` en algunos sistemas Linux modernos.

- ¿Qué es la filosofía UNIX?
- ¿Por qué hay quienes consideran que `systemd` va en contra de esta filosofía?
- El libro indica que para ejecutar los *tests* como `root` es necesario que el sistema en el que se está trabajando tenga habilitada como módulo la opción `CONFIG_SCSI_DEBUG`. ¿Qué son `SCSI` y los `scsi_debug` *devices*?
- Liste al menos tres programas de uso común que son instalados por este paquete. Puede incluir los que hemos usado a lo largo del proyecto y en laboratorios.

Instalación de E2fsprogs

E2fsprogs es un paquete con herramientas para manejo de los sistemas de archivos ext2, ext3 y ext4. Durante la instalación de E2fsprogs es necesario actualizar el archivo `dir` para la correcta instalación de su documentación.

- ¿Para qué sirve el archivo `dir` ubicado en `/usr/share/info`?
- ¿Cuál es la diferencia entre un sistema de archivos común y un sistema de archivos que lleva un diario (*journaling file system*)?

Luego de E2fsprogs se instalarán los paquetes Sysklogd y Sysvinit. La instalación de Sysklogd requerirá crear un archivo de configuración en `/etc/syslog.conf`.

- Explique el contenido y propósito de este archivo de configuración.
- Explique los *run levels* y *bootscripts* de SystemV. ¿Qué es SystemV?
- Investigue otro *init system* y explique brevemente las diferencias con SysV Init.

Instalación de Eudev

- Explique brevemente para qué sirve y por qué es importante este paquete. ¿Cuál es la relación entre `udev` y `systemd`?
- Luego de compilar el paquete se ejecutan unas instrucciones que instalan reglas y archivos auxiliares necesarios. ¿Qué son “reglas” y cuál es el propósito de las que se instalan?
- La última instrucción crea la base de datos inicial de dispositivos de *hardware* en el archivo `/etc/udev/hwdb.bin`. ¿Qué información almacena este archivo?

Últimos pasos

Completando el LFS se indica que es necesario identificar el nombre que Udev asigna a los dispositivos de red, en el archivo `/etc/udev/rules.d/70-persistent-net.rules`. Más adelante se requiere la creación de archivos de configuración para la interfaz de red, en el directorio `/etc/sysconfig`.

- ¿Cuál es el nombre de los dispositivos de red? ¿Qué significan los demás datos?
- ¿Cuál es el propósito de este archivo de configuración en el directorio `/etc/sysconfig`?

Configurar el sistema implica crear el archivo `inputrc`, que sirve para configuración del programa Readline instalado previamente.

- ¿Cuál es la importancia del paquete Readline?

Es posible que su LFS presente un error durante el *booteo* similar a lo siguiente:

```
[    1.809393] EXT4-fs error (device sda2): ext4_iget:3898: inode #395882: comm
init: bad extra_isize (55945 != 256)
```

Esto evidencia problemas con el sistema de archivos instalado en la partición de LFS. Para arreglarlo, reinicie su máquina virtual y acceda a su sistema operativo original. Allí, desmonte la partición de *booteo* y la de LFS, y ejecute un chequeo del sistema de archivos con el siguiente comando:

```
sudo fsck -y /dev/sda2
```

La opción `-y` asegura que se apliquen automáticamente todas las reparaciones sugeridas por el programa de chequeo. Esto no es recomendado, pero para este proyecto es efectivo. Una vez completada la revisión y las reparaciones deberá volver a montar los sistemas virtuales de archivos y acceder al ambiente `chroot` (recuerde que la instrucción para hacerlo cambió al concluir, con `Vim`, la instalación de programas del capítulo 6). Desde allí deberá reinstalar los *bootscripts* de LFS.

Al reiniciar su sistema debería poder acceder tanto a su sistema operativo original como a LFS sin problemas. Si es el caso, felicitaciones, ha creado un *Linux From Scratch*.