

UNIVERSIDAD DEL VALLE DE GUATEMALA
Departamento de Ciencias de la Computación
Sistemas Operativos



Excelencia que trasciende

DELVALLE
GRUPO EDUCATIVO

LFS
Preguntas

Davis Alvarez, 15842

GUATEMALA, 27 de noviembre de 2018

Instalación de herramientas necesarias para la creación de un LFS

- ¿Por qué las condiciones de los *if*'s están entre corchetes?
Porque esta es la sintaxis para Linux, dentro de los corchetes se debe poner los parámetros de las condiciones, para que se realice la verificación.
- ¿Cuáles son las condiciones que se están verificando en cada caso?
Se está revisando si los archivos existen y si son un enlace simbólico con `-h`. Mientras que con `-x` se revisa si el archivo existe y si es ejecutable.
- ¿Para qué sirve encerrar una lista de *strings* entre llaves, en este caso?
Sirve para que puede ser recorrida por el `for`, de esta manera es posible su iteración.
- ¿Para qué sirve '\$' en *shell code*?
Puede funcionar de dos formas, la primera para obtener contenido de una variable, al colocarlo antes del nombre. Y también, para ejecutar una subshell, al colocar el símbolo seguido de la subshell entre paréntesis.
- ¿Qué se está haciendo con este ciclo *for*?
Se hace un recorrido y se utilizan todas las librerías de `gmp`, `mpfr` y `mpc`.
- ¿Qué es una *tarball*?
Es un grupo de archivos que se encuentran comprimidos conjuntamente, hace referencia a un archivo `tar`.
- ¿Por qué las *tarball* tienen una extensión “doble” (`.tar.gz`)?
Esto es debido a que primero se encapsula (extensión `.tar`) y luego se comprime (extensión `.gz`, `xz`, `bz2`)

Instalación de `m4` y `bison`

- ¿Qué significa cada una de las opciones (`-xvzf`) utilizadas?
La `x` significa extraer, `v` para verbose, `z` para descomprimir y `f` es utilizado por `tar` para realizar las operaciones.
- ¿Cuál de estas opciones podría haberse omitido?
Se puede omitir `v`, debido a que es una opción extra para el usuario.
- ¿Cuáles son los tres comandos que se deben ejecutar para instalar un paquete en Linux?
Primero se debe ejecutar `./configure` para configurar la instalación, luego `make` para compilar y por último `make install` para instalar.
- ¿Dónde encontramos información adicional de instalación específica para cada paquete?
Generalmente los paquetes traen un archivo `README` con toda esta información.
- ¿Qué es un enlace simbólico?
Es un enlace que apunta hacia otro enlace y no a la dirección de memoria.
- ¿Cuál es la diferencia entre un enlace simbólico (*symlink*) y uno duro (*hard link*)?
Debido a que un enlace simbólico apunta hacia otro enlace y el enlace duro apunta hacia una dirección de memoria. Por esta razón cuando se elimina un enlace duro también se elimina el enlace simbólico.
- En el comando `ln` usado para crear enlaces, ¿en qué orden se deben escribir los parámetros para crear un enlace de *A* a *B*?
El primer parámetro después del `ln`, debe ser el enlace duro, luego se debe poner el simbólico que apunta hacia el primero.

Instalación de herramientas faltantes usando la Advanced Package Tool (`apt`)

- ¿Qué es apt-get y cómo funciona?
- ¿Qué se modificó y se hizo con apt-get mediante las instrucciones anteriores?

Página 14

- ¿Para qué sirve crear un archivo llamado .profile?

Página 15

- ¿Qué es /dev/null?
- Explique las opciones y parámetros usados en este comando.
- ¿Qué significa ser dueño de un directorio?
- ¿Cuál es la diferencia entre una sesión de *login* y una de *no-login* (*login session* vs. *non-login session*)?
- 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`?

Página 16

- `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`?

Primera instalación de Binutils

- ¿Qué es la *target triplet* y cuál es su campo *vendor*?
Describe la plataforma sobre la cual se está ejecutando un script. Este contiene 3 campos, el primero es la familia de CPU, el siguiente es el *vendo*, el cual indica el vendedor de la plataforma, y por último el nombre del sistema operativo.
- ¿Cuál es la importancia de este campo en la *target triplet* y para qué sirve que, en nuestro sistema, tenga el valor `lfs`?
Cuando cambiamos el nombre del *target triplet*, al cambiar el *vendo* nos aseguramos que la primera compilación de Binutils y GCC produzcan un *cross-linker* y *cross-compiler* que sean compatibles.
- ¿Para qué sirve la opción `--prefix=/tools`?
Esta opción le indica al archivo de configuración que prepare la instalación de los programas de Binutils en el directorio especificado.

Primera instalación de GCC

- ¿Qué hacen estos paquetes?
MPFR es una librería de múltiple precisión para el redondeo de punto flotante, GMP es una librería de múltiple precisión para cálculos aritméticos y MPC es una librería de múltiple precisión para C.
- ¿Qué hace el programa `sed`? Aclare con el mayor detalle posible qué está haciendo `sed` con ese *relajo* abajo de `cp -uv $file{.,orig}`.
`Sed` es un editor de flujo de texto el cual transforma y filtra según se especifica. Lo que pasa en la primera instrucción de `sed` es que se está anteponiendo el directorio `/tools` a cada

instancia de `/lib/id`, `/lib64/id` o `/lib32/id`. Mientras que la segunda realiza un reemplazo en las instancias de `/usr` que fueron escritas en el código directamente.

- [¿Qué hace el comando touch?](#)
Sirve para cambiar la hora y fecha de creación, acceso y modificación a archivos o directorios.
- [¿Qué hace la opción `-u` de `cp` y cómo se relaciona con `touch` en este ciclo?](#)
Esto prevee cambios inesperados a los archivos originales. Por si se ejecutan los comandos más de una vez equivocadamente.
- [¿Para qué sirven las macros `STANDARD_STARTFILE_PREFIX_N` y por qué se les asignan los valores que se les asignan en este ciclo?](#)
Estos macros se encuentran configurados, por lo que se deben definir al inicio y luego se les debe asignar una configuración que los dirige hacia el final del archivo.
- [Desglose y explique el funcionamiento de este ciclo.](#)
Primero, se realiza una búsqueda de todos los archivos en el directorio `"gcc/config"` que tienen por nombre `"Linux.h"`, `"Linux.h"` o `"sysv4.h"`. Cada uno de los archivos encontrados es copiado y renombrado con el mismo nombre, más el sufijo `".orig"`. Luego, se ejecuta un `sed` que antepone `"/tools"` a cada instancia de `"/lib/ld"`, `"/lib64/ld"` o `"/lib32/ld"`, mientras que el segundo está reemplazando las instancias de `"/usr"` que fueron escritas en código directamente. Posteriormente, se añaden las instrucciones de `"define"` para alterar el prefijo `"starfile"` por defecto y colocarlo al final de los archivos. Para finalizar, se ejecuta el comando `touch` para actualizar la fecha de creación y acceso de estos nuevos archivos.

Página 19

- [¿Qué es un `sysroot` y por qué se detalla aquí con valor `\$LFS`?](#)
En el `sysroot` se indica cual será la raíz del sistema y se detalla con el valor de la variable `$LFS`, ya que esta es la dirección del sistema que estamos armando.
- [¿Por qué se especifica `\$LFS_TGT` como el valor de `--target`?](#)
De esta manera aseguramos que cambie el target triplet a través del `vendo`, por lo tanto la primera compilación de `Binutils` y `GCC` va a generar un cross-compiler y cross-linker que sean compatibles.
- [¿Qué hace la opción `--disable-nls`?](#)
Con esto se desactiva el soporte para lenguajes nativos, ya que se utilizara ingles como idioma.
- [¿Qué hace la opción `--disable-shared` y por qué es necesaria? ¿Cuál es la diferencia entre *linking* estático y dinámico?](#)
Es utilizado para desligar el sistema que estamos armando del sistema anfitrión (`knoppix`). Lo que fuerza a `GCC` a utilizar sus librerías internas estáticamente. Esto es necesario para que el sistema construido sea independiente. El *linking* estático solamente posee una dirección hacia la cual busca la información, mientras que el *linking* dinámico puede cambiar dinámicamente.

Instalación de Glibc

- [¿Qué hace la ejecución del `script config.guess` en `glibc- 2.21/scripts`?](#)
Sirve para que el programa identifique cual es el sistema sobre el cual se esta compilando.
- [¿Cuál es el efecto de las opciones `--host` y `--build` en este comando?](#)

Esto funciona para que la compilación de Glibc se configure a si misma para que realice cross-compile, utilizando cross-linker y el compilador de /tools

- ¿Qué es *stack unwinding* y qué es *forced stack unwinding*?
Stack unwinding sirve para modificar el stack trace generado por una compilación, indicando las funciones que se están ejecutando. Mientras que *forced stack unwinding* es un permiso especial de este stack para modificar el stack trace.

Página 20

- ¿Qué es un archivo ELF y qué hace la opción `-l` de `readelf`?
- Hasta el momento instalamos GCC y Binutils con configuraciones que conforman lo que el libro llama *cross GCC* y *cross Binutils*. ¿Qué es un *cross compiler*? ¿Por qué instalamos versiones *cross* de GCC y Binutils? 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++/7.2.0`?

Página 21

- ¿Por qué se debe realizar una segunda instalación de Binutils?
 - ¿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` pero provisto por el sistema. ¿Por qué la primera instalación de GCC no hizo esto desde un principio?

Página 22

- `--disable-bootstrap` deshabilita el comportamiento habitual que tiene GCC de compilarse haciendo un *bootstrap build*. ¿Qué es esto?

Página 23

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

Página 24

- ¿Qué es `pkg-config` y por qué se especifica una variable `PKG_CONFIG` vacía en esta configuración?
- ¿Para qué sirve `Ncurses`?
- ¿Qué es `malloc`?
- ¿Para qué sirve `M4`?
- ¿Qué hace el programa `hostname` de Coreutils?

Página 25

- ¿Cuál es la relación entre los programas msgfmt, msgmerge y xgettext de Gettext?
- ¿Qué es GNU Guile?
- ¿Cuál es la relación entre Patch y Diffutils?

Página 26

- ¿Qué es la filosofía UNIX?
- ¿Por qué hay quienes consideran que systemd va en contra de esta filosofía?
 - ¿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?

Página 27

- ¿Qué significa “montar” un sistema de archivos?
 - ¿Qué es un sistema de archivos? ¿Qué es un sistema de archivos virtual?
 - Explique el propósito de cada uno de los sistemas de archivos /dev, /proc, /sysfs y /run.
 - Explique a detalle qué hacen los comandos del inciso *b*.
 - Explique a detalle qué hacen los comandos mount de este inciso.
 - Explique exactamente qué hace el comando anterior y por qué es importante para el resto del proyecto.

Página 29

- ¿Por qué es necesario crear ese segundo conjunto de *links* simbólicos que apuntan a elementos en /tools/bin?
- ¿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)?

Página 30

- ¿Para qué sirven los archivos /etc/passwd y /etc/group? Explique el formato de estos archivos.
- Explique el propósito y configuración de los usuarios creados en /etc/passwd.
- ¿Cuál es el efecto visible del comando ejecutado en este inciso, y por qué se ejecuta usando exec en lugar de sólo /tools/bin/bash?

Página 31

- Dé un vistazo al contenido del archivo glibc-2.26-fhs-1.patch, y responda: ¿qué hace el programa patch?
- ¿Para qué sirven las opciones -Np1 e -i utilizadas?
- ¿Qué hace touch cuando recibe un archivo inexistente como argumento?

Página 32

- ¿Qué es nscd?
- Independientemente del camino que haya tomado responda: ¿qué es un locale?

Página 33

- ¿Cuál es el propósito de nsswitch.conf?
- ¿Cuál es el propósito de cada una de las bases de datos configuradas en este archivo en el comando anterior?
- ¿Para qué sirve el comando zic?
- ¿Cuál es el propósito de los archivos zone.tab, zone1970.tab e iso3166.tab?

Página 34

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

Página 35

- ¿Qué hace la opción -Wl en el comando cc?
- ¿Para qué sirve usar &>?
- ¿Cuál es el efecto de los símbolos ., [] y * en los argumentos de grep?
- ¿Qué hace la opción -o de grep?

Página 36

- Explique el propósito de los archivos crt1.o, crt1.o y crtn.o
- En el comando grep con la opción -B1, ¿qué sucedería si no usamos esa opción?
- De acuerdo a 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*?

Página 38

- Investigue:
 - o ¿En qué consiste una PTY?
 - o ¿Cómo funcionan las PTY?
 - o ¿Cómo se relacionan con expect?
 - o ¿Cómo se relacionan con el sistema de archivos devpts que montamos casi al inicio de la construcción del LFS?

Página 39

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

Página 41

- ¿Qué es y para qué sirve LTO?

Página 43

- Explique el propósito y funcionamiento de `\1` en el comando anterior.
- ¿Para qué sirve `Pkg-config`?

Página 44

- 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?

Página 45

- ¿Cuál es la diferencia entre `Curses` y `Ncurses`?
- Investigue qué son las *automatic variables* en un *makefile* y explique el funcionamiento de los últimos dos comandos `make`.

Página 46

- ¿Para qué sirve este paquete? Incluya una breve explicación de qué son los *extended attributes* en un sistema de archivos.
- ¿Qué son las listas de control de acceso (ACL's)?

Página 47

- Explique la relación entre listas de control de acceso y *capabilities*.
- ¿Para qué sirve `Shadow`?

Página 48

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

Página 49

- ¿Cuál es la diferencia entre `Lex` y `Flex`?

Página 50

- ¿Cuál es el propósito/beneficio que provee el *GNU library support script*?

Página 51

- ¿Qué es un *perfect hash*?

Página 52

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

Página 53

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

Página 55

- ¿Qué es un dispositivo `tty`?

Página 56

- ¿Para qué sirve el archivo `dir` ubicado en `/usr/share/info`?

Página 57

- ¿Cuál es la diferencia entre un sistema de archivos común y un sistema de archivos que lleva un diario (*journaling file system*)?

Página 58

- Gawk es la implementación GNU de Awk. Describa Awk.

Página 59

- ¿Cuál es la importancia de Groff (y sus predecesores) en el contexto de los sistemas Unix y sus derivados?

Página 61

- Explique el contenido y propósito de este archivo de configuración.

Página 62

- Explique los *run levels* de SystemV.

- Investigue otro *init system*, y explique brevemente las diferencias con SysV Init.

Pagina 63

-
- Explique brevemente para qué sirve y por qué es importante este paquete. ¿Cuál es la relación entre udev y systemd?
-
- Las primeras instrucciones instalan algunas 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?
-
- ¿Qué hacen el archivo de configuración `adjtime` y el programa `hwclock`?
-

Pagina 64

- El libro indica que para poder 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.

Pagina 65

- Explique brevemente la diferencia entre *man pages* e *info pages*.
- ¿Qué es `vimrc`?
- ¿Qué ventajas presenta Vim sobre Vi?

Página 67

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

Página 69

- ¿Cómo funcionan los *run levels* de init?

Pagina 71

- ¿Cuál es la importancia del paquete `Readline`?

