

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?
apt-get es un paquete que ofrece una forma sencilla de instalar paquetes desde la línea de órdenes. El cual no intenta comprender los ficheros, sino que funciona con el nombre real del paquete y sólo puede instalar ficheros desde una fuente.
- ¿Qué se modificó y se hizo con apt-get mediante las instrucciones anteriores?
Lo que se modificó fue que realizara la instalación de los paquetes según su nombre y la versión especificada, de esta manera logramos instalar las versiones que necesitamos para construir el LFS.
- ¿Para qué sirve crear un archivo llamado .profile?
Este archivo es útil para indicar la partición que se utilizara para cargar el LFS y para guardar los archivos del sistema.
- ¿Qué es /dev/null?
Acá es donde se almacenan los errores que se producen en ejecución, en donde son lanzados a un contenedor basura sin posibilidad de recuperación.
- Explique las opciones y parámetros usados en este comando.
En el comando la opción k sirve para indicar la utilización de una key, para el usuario del LFS que le concede los mismos permisos que el usuario principal.
- ¿Qué significa ser dueño de un directorio?
Esto significa que al ser el dueño puede tener control total sobre el directorio, por lo que no requiere permisos al realizar una acción.
- ¿Cuál es la diferencia entre una sesión de login y una de no-login (login session vs. non-login session)?
La diferencia es que en una sesión sin login se utiliza al querer un espacio o un directorio sin permisos. Mientras que en la sesión login ya se cuenta con un usuario con permisos especificados.
- Explique exactamente qué se está haciendo con env -i HOME=\$HOME TERM=\$TERM PS1='u:\w\\$ ' /bin/bash.
En esta instrucción lo que se hace es que se está copiando las variables del sistema hacia el usuario que se ha creado.
- ¿Cuál es el efecto o propósito de preceder el comando anterior con exec?
Este comando se ejecuta para que sea posible ejecutar los subprocessos siguientes.
- set +h apaga la funcionalidad de hashing para la shell. ¿Para qué sirve el hashing de la shell?
Este comando evita la búsqueda de la variable PATH al guardar su valor en memoria.
- ¿Qué hace el comando umask y cuál es el efecto de usarlo con el parámetro 022?
Este comando sirve para otorgarle permisos de lectura y escritura a un archivo. Al utilizar el parámetro 022 se está indicando que el archivo va a poseer los permisos por defecto.

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 vendor, 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 vendor 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 $file{,orig}`.
`Sed` es un editor de flujo de texto el cual transforma y filtra según se especifica. Lo que pasa en la primer instrucción de `sed` es que se está anteponiendo el directorio `/tools` a cada instancia de `/lib/ld`, `/lib64/ld` o `/lib32/ld`. Mientras que la segunda realiza un reemplazo en las instancias de `/usr` que fueron escritas en el código directamente.
- ¿Qué hace el comando `touch`?
Sirve para cambiar la hora y fecha de creación, acceso y modificación a archivos o directorios.
- ¿Qué hace la opción `-u` de `cp` y cómo se relaciona con `touch` en este ciclo?
Esto previene cambios inesperados a los archivos originales. Por si se ejecutan los comandos más de una vez equivocadamente.
- ¿Para qué sirven las macros `STANDARD_STARTFILE_PREFIX_N` y por qué se les asignan los valores que se les asignan en este ciclo?
Estos macros se encuentran configurados, por lo que se deben definir al inicio y luego se les debe asignar una configuración que los dirige hacia el final del archivo.
- Desglose y explique el funcionamiento de este ciclo.
Primero, se realiza una búsqueda de todos los archivos en el directorio `"gcc/config"` que tienen por nombre `"Linux.h"`, `"Linux.h"` o `"sysv4.h"`. Cada uno de los archivos encontrados es copiado y renombrado con el mismo nombre, más el sufijo `".orig"`. Luego, se ejecuta un `sed` que antepone `"/tools"` a cada instancia de `"/lib/ld"`, `"/lib64/ld"` o `"/lib32/ld"`, mientras que el segundo está reemplazando las instancias de `"/usr"` que fueron escritas en código directamente. Posteriormente, se añaden las instrucciones de `"define"` para alterar el prefijo `"startfile"` por defecto y colocarlo al final de los archivos. Para finalizar, se ejecuta el comando `touch` para actualizar la fecha de creación y acceso de estos nuevos archivos.
- ¿Qué es un *sysroot* y por qué se detalla aquí con valor `$LFS`?
En el `sysroot` se indica cuál 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 tripiet a través del vendó, 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.
- ¿Qué es un archivo ELF y qué hace la opción `-l` de `readelf`?
Este es un archivo con un formato de ejecución de los sistemas Linux. La opción `i` de `readelf` es para desplegar información contenida en el encabezado de este archivo.
- 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.
Un cross-compiler es un compilador que compila para una plataforma ajena, es decir sobre la que se esta ejecutando la compilación. Esto significa que compila algo que el host no puede ejecutar. Se instalan versiones cross para GCC y Binutils debido a que de esta forma se producen archivos binarios compatibles para otras arquitecturas.
- ¿Cuál es el efecto de compilar con `$LFS_TGT-gcc` en lugar de sólo `gcc`?
Se asegura que utilice la librería GCC ubicada dentro del sistema que se esta desarrollando y no la de knoppix.
- Explique la relación entre Binutils, GCC, Glibc y Libstdc++.
Binutils maneja links y archivos ejecutables y a su vez contiene un ensamblador. GCC contiene la colección de compilación de GNU que incluye los compiladores de C y C++. Por ultimo Libstdc++ contiene las librerías utilizadas por c y c++.
- ¿Por qué se usa la opción `--with-gxx-include-dir=/tools/$LFS_TGT/include/c++/7.2.0`?
Esta es la dirección en donde estan ubicadas las librerías estandar include, las cuales necesita el compilador de c ++. Ya que al realizar una compilación esta información es dada por Libstdc++.

Segunda instalación de Binutils

- ¿Por qué se debe realizar una segunda instalación de Binutils?
Esto se hace para activar la nueva configuración, la cual permite añadir el path.
- ¿Para qué sirve el comando `dirname`?
Este comando devuelve un string con el path hasta la ultima carpeta y el ultimo “/”
- ¿Cuál es la diferencia entre ejecutar algo con `$` y ejecutarlo con ```?
La diferencia es que al utilizar `$` se sustituye el comando, pero utilizando el nuevo comando como un parámetro adicional. Mientras que si se utiliza ``` se está diciendo que se requiere una sustitución de comando sin importar cuando sea utilizado.
- ¿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?
El archivo `limits.h` especifica las propiedades de los tipos de archivos que se van a utilizar. Lo sucedido es que se ha creado una referencia hacia este archivo. Por esta razón en este momento se realiza la instalación completa, ya que anteriormente fue instalado parcialmente sin la totalidad de sus funcionalidades.
- `--disable-bootstrap` deshabilita el comportamiento habitual que tiene GCC de compilarse haciendo un *bootstrap build*. ¿Qué es esto?
El Bootstrap build sirve para hacer el sistema GCC se compila a si mismo completamente y que realice una verificación. Al realizar este tipo de compilación se tiene un sistema más completo y de mejor rendimiento.
- ¿Qué es Tcl y con qué propósito fue creado?
Tcl es un paquete que contiene un lenguaje de comandos Tool, este fue creado como prototipo rápido, para aplicaciones en scripts, GUI, y tests.
- ¿Para qué sirve Expect?
Este es un paquete que contiene un script que lleva diálogos programados hacia otros scripts, por lo que de esta forma permite comunicarse con otros programas.
- ¿Cuál es la relación entre DejaGNU, Expect y Tcl?
La relación es que estos 3 paquetes son instalados para brindar soporte en la realización de pruebas para GCC y Binutils.
- ¿Qué es `pkg-config` y por qué se especifica una variable `PKG_CONFIG` vacía en esta configuración?
Este paquete contiene meta información sobre las librerías instaladas. La variable `PKG_CONFIG` se deja vacía para indicar que la configuración ignore cualquier opción relacionada con `pkg-config`, lo cual puede causar que la instalación realice enlaces hacia librerías que están fuera del directorio `/tools`
- ¿Para qué sirve `Ncurses`?
Este paquete tiene librerías útiles para el manejo de pantallas de caracteres sin importar la terminal utilizada.
- ¿Qué es `malloc`?

Esto es una función conocida por causar segmentation faults, de ahí su nombre Bash's memory allocation.

- [¿Para qué sirve M4?](#)
Este es un paquete que contiene macros. Por lo que realiza copias de una entrada a una salida expandiendo macros en el proceso. Cada macro puede estar definido por defecto o pueden ser definidos por el usuario.
- [¿Qué hace el programa hostname de Coreutils?](#)
Este comando devuelve el nombre del sistema anfitrión.
- [¿Cuál es la relación entre los programas msgfmt, msgmerge y xgettext de Gettext?](#)
La relación es que cada uno de estos programas realizan tareas secuenciales para generar una plantilla de traducción. Msgfmt realiza un mensaje en binario desde un catálogo de traducción, msgmerge sirve para combinar dos traducciones abiertas en un solo archivo y xgettext sirve para extraer los mensajes a traducir desde los archivos de código fuente.
- [¿Qué es GNU Guile?](#)
Es una librería que fue diseñada para ayudar a los programadores en la creación de aplicaciones más flexibles, en donde se extendió su funcionalidad a través de la utilización de plugins, módulos o scripts.
- [¿Cuál es la relación entre Patch y Diffutils?](#)
La relación es que Diffutils contiene programas que muestran las diferencias entre archivos y directorios. Y patch puede modificar o crear archivos aplicando un parche, generalmente creado por Diffutils.
- [¿Qué es la filosofía UNIX?](#)
Es un conjunto de directrices para el desarrollo de un sistema operativo pequeño, con alta capacidad y con una interfaz simple y limpia, el cual fue creado por Ken Thompson.
- [¿Por qué hay quienes consideran que systemd va en contra de esta filosofía?](#)
Esto se debe a que esta instrucción integra otras funcionalidades que no son básicas para el inicio de un sistema. Por esta razón se considera que va contra la filosofía UNIX, ya que carga el inicio del sistema de manera innecesaria.
- [¿Qué son *debugging symbols*?](#)
Estos contienen información que indica cuál es el lenguaje de programación de un script que genere un segmento de código de máquina en un módulo ejecutable.
- [¿Para qué sirve strip?](#)
Este sirve para destacar símbolos de archivos de carácter ejecutable.
- [¿Por qué el primer comando simplemente ejecuta strip, pero el segundo ejecuta /usr/bin/strip?](#)
El primer comando es para el sistema host, el cual se ejecuta de esta manera para guardar los archivos binarios en el directorio especificado. Mientras que en el segundo ya no es necesario.
- [Explique el concepto de *relocation* asociado a los *linkers*, y su relación con la opción `--strip-unneeded`?](#)
Relocation es utilizado para hacer una unión entre las referencias simbólicas con definiciones simbólicas. Mientras que la función strip unneeded sirve para indicar que no se deben destacar los símbolos.

Construcción de LFS (parte I)

- **Qué significa “montar” un sistema de archivos?**
Esto he referencia a realizar la asignación de espacio en memoria y aplicar la estructura utilizada por el sistema de archivos, así el sistema operativo es capas de hacer uso de esta.
- **¿Qué es un sistema de archivos? ¿Qué es un sistema de archivos virtual?**
Un sistema de archivos virtual es una capa de abstracción de los sistemas de archivos comunes y su principal propósito es permitir que las aplicaciones cliente tengan acceso a diversos tipos de sistemas de archivos concretos de una manera uniforme.
Un sistema de archivos es la parte del sistema operativo que se encarga de administrar y facilitar el uso de memoria secundaria. Una de sus funciones principales es administrar el espacio disponible, asignar un espacio a un archivo que lo solicite y acceder a los datos que contiene guardados.
- **Explique el propósito de cada uno de los sistemas de archivos /dev, /proc, /sysfs y /run.**
/proc es un directorio virtual que contiene vistas dentro del kernel, /dev es una carpeta que contiene archivos disponibles para el sistema, /sysfs es un directorio que provee de significado a las estructuras exportadas del kernel, y /run es un directorio en el que están disponibles y se pueden editar, y contiene las herramientas para el inicio del sistema.
- **Explique a detalle qué hacen los comandos del inciso b.**
Lo que se esta realizando es que los nodos especificados se encuentren disponibles antes de que inicie udevd y durante sus ejecución, por lo que Linux empezara con la dirección especifica en el init.
- **Explique a detalle qué hacen los comandos mount de este inciso.**
Se realiza el mount de los sistemas de archivos ya especificados. En donde el parámetro v muestra el tipo de versión que es el archivo y t lista todos los sistemas de ficheros ya montados.
- **Explique exactamente qué hace el comando anterior y por qué es importante para el resto del proyecto.**
Este es una llamada del sistema que realiza un cambio a la localidad de directorio raíz temporalmente al folder especificado. La importancia de esto radica en que al especificar el nuevo directorio se pueden realizar instalaciones sin modificar el sistema operativo host.
- **¿Por qué es necesario crear ese segundo conjunto de *links* simbólicos que apuntan a elementos en /tools/bin?**
Esto es necesario ya que algunos programas no están instalados. Por esto se crean links simbólicos que luego serán reemplazados con los archivos de la instalación.
- **¿Para qué sirve el comando install?**
Este comando facilita la instalación de paquetes, al copiar los archivos, cambiar el propietario y otorgarle permisos.
- **¿Para qué sirve el archivo /etc/mstab, y por qué se *linkea* a /proc/self/mounts?**
Este archivo donde Linux mantiene una lista de archivos montados. Este archivo será creado al montar el sistema desarrollado, por lo que se linkea a este directorio para luego ser creado.
- **¿Qué es el *Filesystem Hierarchy Standard* (FHS)?**
Este es el que define la estructura y los directorios de contenido en Linux.
- **¿Para qué sirven los archivos /etc/passwd y /etc/group? Explique el formato de estos archivos.**

Estos directorios son los que determinan las contraseñas de los usuarios raíces y los grupos, los cuales serán especificados más adelante.

- [Explique el propósito y configuración de los usuarios creados en /etc/passwd.](#)
El propósito de este archivo es listar las cuentas que existen en el sistema, especificando la información de cada usuario. Este archivo contiene una entrada por línea para cada usuario, en donde se especifican: nombre de usuario, contraseña, id del usuario, id del grupo, información extra del usuario, directorio home, línea de comandos.
- [¿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?](#)
Para quitar el nombre “i had no name” se ejecuta una nueva terminal. Y gracias a que ya se instala Glibc y se configuro las contraseñas y grupos del usuario, ya funcionara esta nueva terminal.

Instalación de herramientas en ambiente aislado

- [Dé un vistazo al contenido del archivo glibc-2.26-fhs-1.patch, y responda: ¿qué hace el programa patch?](#)
Este programa elimina las referencias al directorio var/db que no es parte del FHS y realiza un reemplazo por directorios que si nos sirven, al reemplazar /var/db/nscd por /var/cache/nscd; y /var/db/ lo reemplaza por /var/lib/nss_db/.
- [¿Para qué sirven las opciones -Np1 e -i utilizadas?](#)
Np1 ignora los parches que patch piensa que están invertido o que ya están aplicados, con nombre de archivo de la franja y i lee el parche del archivo parche en lugar de la entrada estándar.
- [¿Qué hace touch cuando recibe un archivo inexistente como argumento?](#)
Al recibir un parámetro inesperado crea el archivo.
- [¿Qué es nscd?](#)
Esto es el Name Service Cache Daemon, el cual es un demonio que provee una caché para las solicitudes de los servicios mas comunes en el sistema.
- [Independientemente del camino que haya tomado responda: ¿qué es un locale?](#)
Locale es lo que se utiliza al definir el idioma que se utilizara en el sistema, por lo que aquí se especifican los caracteres utilizados por el usuario.
- [¿Cuál es el propósito de nsswitch.conf?](#)
Esto es necesario debido a que aunque Glibc provee acciones predeterminadas para cuando no hay un archivo o es un archivo es corrupto, estas acciones predeterminada no funcionan del todo bien en el entorno red.
- [¿Cuál es el propósito de cada una de las bases de datos configuradas en este archivo en el comando anterior?](#)
Es proveer un archivo con la especificación de como debe funcionar el proceso de búsqueda de estos servicios.
- [¿Para qué sirve el comando zic?](#)
Este es un compilador de zona horaria, el cual lee el texto desde un archivo nombrado en línea de comandos y crea los archivos de información de la conversión de tiempo especificado en dicha entrada.
- [¿Cuál es el propósito de los archivos zone.tab, zone1970.tab e iso3166.tab?](#)

Este archivo contiene una tabla con los códigos de los países, con latitud, longitud, valores TZ y comentarios descriptivos para la zona horario. Por lo que estos archivos con los que describen el formato de las distintas zonas horarias junto con toda la información que representa cada una de ellas.

- [¿Qué hace la instrucción `n` en `sed`?](#)
Esta instrucción imprime el espacio del patrón actual, lo elimina y lee la siguiente línea del input.
- [¿Cuál es el propósito del archivo `specs`?](#)
Este comando sirve para apuntar al nuevo link dinámico, al eliminar todas las instancias de `/tools` y dejando el path correcto. Además esto ajusta el archivo para que GCC sepa dónde buscar los headers y archivo de Glibc.
- [¿Qué hace la opción `-Wl` en el comando `cc`?](#)
Se le pasa el `verbose` como opción al linker para que la agregue al ejecutable cuando este sea creado.
- [¿Para qué sirve usar `&>`?](#)
Esto redirige el output del comando hacia el archivo especificado.
- [¿Cuál es el efecto de los símbolos `.`, `\[\]` y `*` en los argumentos de `grep`?](#)
Estos parámetros ayudan a delimitar la búsqueda a través de una expresión regular.
- [¿Qué hace la opción `-o` de `grep`?](#)
Esto funciona para imprimir solamente la parte que coincide con la expresión regular que se usó, sin imprimir toda la línea.
- [Explique el propósito de los archivos `crt1.o`, `crti.o` y `crtn.o`](#)
`Crtn.o` define la función `epilog`, `crt1.o` contiene el símbolo iniciado que se encarga de preparar el ambiente con `argc/argv/libc_init/libc_fini` antes de entrar a `Libc`, `crti.o` define la función `prolog_init` en la sección `init` y `_fini` en la sección `.fini`.
- [En el comando `grep` con la opción `-B1`, ¿qué sucedería si no usamos esa opción?](#)
Esta es utilizada para separación entre contextos, pues esto imprime una línea con el contexto superior entre cada grupo de coincidencias.
- [De acuerdo a estos resultados, ¿dónde se inicia la búsqueda de encabezados que #incluimos en nuestros programas?](#)
Esto inicia la búsqueda en `/usr/include`.
- [¿Qué es, y para qué sirve un `soname`? Explique qué son `libc.so.6` y `ld-linux.so.2` \(se recomienda investigar el concepto de *shared library*\).](#)
Este es un campo que contiene la información en `shared object file`, el cual es un string que describe la funcionalidad del objeto. El cual es utilizado generalmente para ofrecer compatibilidad hacia atrás.
- [¿Qué es *underlinking*?](#)
Esto es utilizado en la situación en donde un binario utiliza símbolos que no están provistos por las librerías a las que está linkado.

Investigue:

- [¿En qué consiste una PTY? ¿Cómo funcionan las PTY?](#)
Este es un par de pseudo-dispositivos donde un esclavo simula una terminal y el maestro provee lo que necesita para que simulador de terminal sea controlado.
- [¿Cómo se relacionan con `expect`?](#)
Se relaciona en que `expect` utiliza el proceso de simulación de la terminal para interactuar con el usuario y alimentar de texto al master.

- ¿Cómo se relacionan con el sistema de archivos devpts que montamos casi al inicio de la construcción del LFS?

Se relacionan en que con PTY se simula una terminal. Ya que Linux crea una PTY para una nueva terminal que abre el usuario y muestra la entrada correspondiente en /dev/pts. El dispositivo PTY actúa como terminal y acepta entradas desde el teclado y luego muestra las salidas en texto.

- ¿Qué hace la opción -k?

La opción k permite al compilador continuar sin ponerle atención a los errores, por lo que se ejecuta lo más posible hasta que encuentra un error crítico y no puede continuar con el proceso de compilación.