Para Gente Nueva en FreeBSD y UNIX®

Resumen

¡Enhorabuena por instalar FreeBSD! Esta introducción es para gente nueva en FreeBSD *y* en UNIX® - así que se comienza con lo básico.

Tabla de contenidos

1. Iniciar sesión y salir . . . 1
2. Agregar un Usuario con Privilegios de Root . . . 2
3. Echando un vistazo . . . 3
4. Obteniendo ayuda e información . . . 4
5. Editando texto . . . 5
6. Otros comandos útiles . . . 7
7. Próximos pasos . . . 7
8. Tu entorno de trabajo . . . 8
9. Otros . . . 10
10. Comentarios Bienvenidos . . . 10

1. Iniciar sesión y salir

Inicia sesión (cuando ves `login:`) como un usuario que has creado durante la instalación o como `root`. (Tu instalación de FreeBSD ya tendrá una cuenta para `root`; quien puede ir a cualquier sitio y hacer de todo, incluyendo borrar ficheros esenciales, así que ¡ten cuidado!) Los símbolos % y # en lo que sigue significan prompt (el tuyo podría ser diferente), con % indicando un usuario ordinario y # para indicar `root`.

Para cerrar sesión (y obtener un nuevo prompt `login:`) teclea

```
# exit
```

tantas veces como sea necesario. Sí, presiona `enter` después de los comandos y recuerda que UNIX® distingue entre mayúsculas y minúsculas - `exit`, no `EXIT`.

Para apagar el ordenador, escribe

```
# /sbin/shutdown -h now
```

O para reiniciar, escribe

```
# /sbin/shutdown -r now
```

o

```
# /sbin/reboot
```

También puedes reiniciar con `Ctrl` + `Alt` + `Delete`. Dale un poco de tiempo para que haga su trabajo. Es equivalente a `/sbin/reboot` en versiones recientes de FreeBSD y es mucho, mucho mejor que presionar el botón de reset. No quieres tener que reinstalar esto ¿verdad?

2. Agregar un Usuario con Privilegios de Root

Si no creaste un usuario cuando instalaste el sistema y por lo tanto has iniciado sesión como `root`, probablemente deberías crear uno ahora con

```
# adduser
```

La primera vez que usas `adduser`, podría preguntar por algunos valores por defecto para guardarlos. Podrías querer hacer que el shell por defecto fuera csh(1), si sugiere que el valor por defecto es `sh`. De lo contrario, simplemente presiona enter para aceptar cada valor por defecto. Estos valores por defecto se salvan en /etc/adduser.conf, un fichero que se puede editar.

Imagina que creas un usuario `jack` con nombre completo *Jack Benimble*. Asigna a `jack` una contraseña si la seguridad (incluso si son niños cerca que podrían acceder al teclado) es un problema. Cuando te pregunte si quieres invitar a `jack` a otros grupos, teclea `wheel`

```
Login group is "jack". Invite jack into other groups: wheel
```

Esto hará posible iniciar sesión como `jack` y usar el comando su(1) para convertirse en `root`. Después ya no serás reprendido nunca más por iniciar sesión como `root`.

Puedes salir de `adduser` en cualquier momento tecleando `Ctrl` + `C`, y al final tendrás la oportunidad de aprobar el nuevo usuario o simplemente teclear `n` para no hacerlo. Podrías querer crear un segundo usuario de forma que cuando edites los ficheros de inicio de sesión de `jack` tengas un repuesto en caso de que algo salga mal.

Una vez hecho esto, utiliza `exit` para volver al prompt de login e inicia sesión como `jack`. En general, es una buena idea hacer todo el trabajo que sea posible como un usuario ordinario que no tiene el poder - y el riesgo - de `root`.

Si ya has creado un usuario y quieres que el usuario sea capaz de hacer `su` a `root`, puedes iniciar

sesión como `root` y editar el fichero /etc/group, añadiendo `jack` a la primera línea (el grupo `wheel`). Pero primero necesitas practicar con vi(1), el editor de texto - o usa el editor de texto ee(1), más sencillo y que viene instalado en versiones recientes de FreeBSD.

Para eliminar un usuario, utiliza `rmuser`.

3. Echando un vistazo

Inicie sesión como un usuario normal, eche un vistazo y pruebe algunos comandos que accederán a las fuentes de ayuda e información de FreeBSD.

Aquí se describen algunos comandos y lo que hacen:

`id`

¡Te dice quién eres!

`pwd`

Te muestra dónde estás—el directorio de trabajo actual.

`ls`

Lista los archivos en el directorio actual.

`ls -F`

Lista los ficheros en el directorio actual con un * después de los ficheros ejecutables, un `/` después de los directorios, y una `@` después de los enlaces simbólicos.

`ls -l`

Muestra los archivos en formato largo: tamaño, fecha, permisos.

`ls -a`

Lista ficheros ocultos "dot" junto a los demás. Si eres `root`, los ficheros "dot" se muestran sin necesidad de usar la opción `-a`.

`cd`

Cambia directorios. `cd ..` vuelve hacia atrás un nivel; fíjate en el espacio después de `cd`. `cd /usr/local` va a ese directorio. `cd ~` va al directorio home del usuario que ha iniciado sesión, por ejemplo, /usr/home/jack. Prueba `cd /cdrom`, y luego `ls`, para ver si tu CDROM está montado y funcionando.

`less filename`

Te permite ver el fichero (llamado *filename*) sin cambiarlo. Prueba `less /etc/fstab`. Teclea `q` para salir.

`cat filename`

Muestra *filename* por pantalla. Si es muy largo y sólo puedes ver la última parte, presiona `ScrollLock` y utiliza `up-arrow` para moverte hacia atrás; puedes usar `ScrollLock` también con páginas del manual. Presiona `ScrollLock` de nuevo para salir. Podrías querer probar `cat` en algunos de los ficheros dot en tu directorio home-`cat .cshrc`, `cat .login`, `cat .profile`.

Notarás que hay alias en .cshrc para algunos de los comandos `ls` (son muy útiles). Puedes crear otros alias editando .cshrc. Puedes poner estos alias disponibles para todos los usuarios del sistema poniéndolos en el fichero de configuración de `csh` a nivel de sistema, en /etc/csh.cshrc.

4. Obteniendo ayuda e información

Aquí hay algunas fuentes de ayuda. *Text* significa algo que escojas tú y que teclees-normalmente un comando o un nombre de fichero.

`apropos text`

Todo lo que contenta la cadena *text* en la `base de datos whatis`.

`man text`

La página del manual para *text*. La mayor fuente de documentación de los sistemas UNIX®. `man ls` te dirá de qué forma se puede usar `ls`. Presiona `Enter` para moverte por el texto, `Ctrl` + `B` para volver atrás una página, `Ctrl` + `F` para avanzar, `q` o `Ctrl` + `C` para salir.

`which text`

Te dice dónde se encuentra el comando *text* en el path del usuario.

`locate text`

Todas las rutas donde se encuentra la cadena *text*.

`whatis text`

Te dice lo que hace el comando *text* y cuál es su página de manual. Teclear `whatis *` te dará información acerca de todos los binarios en el directorio actual.

`whereis text`

Encuentra el fichero *text*, devolviendo su ruta completa.

Podrías querer probar a usar `whatis` en algunos comandos útiles y comunes como `cat`, `more`, `grep`, `mv`, `find`, `tar`, `chmod`, `chown`, `date`, and `script`. `more` te permite leer una página cada vez como hace en DOS, por ejemplo, `ls -l | more` o `more filename`. El * funciona como un wildcard, por ejemplo, `ls w*` mostrará todos los ficheros que empiezan por `w`.

¿Algunos de estos no funcionan muy bien? Tanto locate(1) como whatis(1) dependen de una base de datos que se reconstruye semanalmente. Si tu máquina no va a estar encendida durante el fin de semana (y ejecutando FreeBSD), podrías querer ejecutar estos comandos de mantenimiento diariamente, semanalmente, y mensualmente de vez en cuando. Ejecútalos como `root` y, de momento, dale tiempo a que termine cada uno antes de empezar con el siguiente.

```
# periodic daily
output omitted
# periodic weekly
output omitted
# periodic monthly
output omitted
```

Si te cansas de esperar, presiona `Alt` + `F2` para ir a otra *consola virtual* e iniciar sesión de nuevo. Después de todo, es un sistema multiusuario y multitarea. De todas formas estos comandos probablemente mostrarán mensajes en la pantalla mientas se ejecutan; puedes teclear `clear` en el prompt para limpiar la pantalla. Una vez que han terminado, podrías mirar en /var/mail/root y /var/log/messages.

Ejecutar dichos comandos es parte de la administración del sistema-y como único usuario de un sistema UNIX®, tú eres el único administrador. Virtualmente todo lo que necesitas hacer como `root` es administración del sistema. Estas responsabilidades no están bien cubiertas incluso en esos libros gordos de UNIX®, que parecen dedicar mucho espacio a desplegar menús en gestores de ventanas. Podrías querer obtener uno de los dos libros líderes en administración de sistemas, bien Evi Nemeth et.al.'s UNIX System Administration Handbook (Prentice-Hall, 1995, ISBN 0-13-15051-7)-la segunda edición con la cubierta roja; o Æleen Frisch's Essential System Administration (O'Reilly & Associates, 2002, ISBN 0-596-00343-9). Yo uso Nemeth.

5. Editando texto

Para configurar tu sistema, necesitas editar ficheros de texto. La mayoría de ellos estarán en el directorio /etc; y necesitarás hacer `su` a `root` para poder cambiarlos. Puedes usar el sencillo `ee`, pero a la larga merece la pena aprender `vi`. Hay un tutorial excelente de vi en /usr/src/contrib/nvi/docs/tutorial, si tienes instaladas los fuentes del sistema.

Antes de editar un fichero, probablemente deberías hacer una copia de seguridad. Imagina que quieres editar /etc/rc.conf. podrías utilizar `cd /etc` para ir al directorio /etc y hacer:

```
# cp rc.conf rc.conf.orig
```

Esto copiaría rc.conf a rc.conf.orig, y después copiarías rc.conf.orig a rc.conf para recuperar el original. Pero sería incluso mejor mover (renombrar) y luego copiarlo de vuelta:

```
# mv rc.conf rc.conf.orig
# cp rc.conf.orig rc.conf
```

porque `mv` conserva la fecha y propietario originales del fichero. Ahora ya puedes editar rc.conf. Si quieres recuperar el original, harías `mv rc.conf rc.conf.myedit` (asumiendo que quieres mantener la versión editada) y luego

```
# mv rc.conf.orig rc.conf
```

para dejar las cosas como estaban.

Para editar un fichero, escribe

```
# vi filename
```

Muévete por el texto con las teclas de dirección. `Esc` (la tecla de escape) pone a `vi` en modo comando. Aquí hay algunos comandos:

`x`
: borra la letra que se encuentre en la posición del cursor

`dd`
: elimina la línea entera (incluso si no aparece por completo en la pantalla)

`i`
: inserta texto en la posición del cursor

`a`
: inserta texto después del cursor

Una vez que tecleas `i` o `a`, ya puedes introducir texto. `Esc` te lleva de vuelta all modo comando donde puedes teclear

`:w`
: para guardar los cambios en el disco y continuar con la edición

`:wq`
: para grabar y salir

`:q!`
: para salir sin grabar los cambios

`/text`
: para mover el cursor a *text*; `/` `Enter` (la tecla enter) para encontrar la siguiente instancia de *text*.

`G`
: para ir al final del archivo

`nG`
: para ir a la línea *n* en el fichero, donde *n* es un número

`Ctrl` + `L`
: para recargar la pantalla

`Ctrl` + `b` y `Ctrl` + `f`
: ir hacia atrás y hacia adelante una pantalla, como se hace con `more` y `view`.

Practica con `vi` en tu directorio home creando un nuevo fichero con `vi filename` y añadiendo y borrando texto, guardando el fichero y abriéndolo de nuevo. `vi` es una caja de sorpresas porque es en realidad bastante complicado y a veces ejecutas un comando sin querer que hará algo que no esperas. (A algunas personas en realidad les gusta `vi` - es más potente que EDIT en DOS- investiga sobre `:r`). Usa `Esc` una o más veces para asegurarte de que estás en modo comando y continúa a partir de ahí cuando tengas problemas, guarda frecuentemente con `:w` y usa `:q!` para salir y empezar de nuevo (desde tu último `:w`) cuando lo necesites.

Ahora ya puedes usar `cd` para moverte a /etc, hacer `su` a `root`, usar `vi` para editar el fichero /etc/group y añadir un usuario al grupo `wheel` de forma que tenga privilegios de root. Simplemente añade una coma y el nombre del usuario al final de la primera línea del fichero, presiona `Esc` y usa `:wq` para escribir el fichero en disco y salir. Efectivo al instante. (No pusiste un espacio después de la coma, ¿verdad?)

6. Otros comandos útiles

`df`
: muestra el espacio en disco y los sistemas de archivos montados.

`ps aux`
: muestra procesos en ejecución. `ps ax` es una forma más compacta.

`rm filename`
: elimina *filename*.

`rm -R dir`
: elimina el directorio *dir* y todos sus subdirectorios-¡cuidado!

`ls -R`
: lista ficheros en el directorio actual y todos los subdirectorios; He usado una variante, `ls -AFR > where.txt` para obtener una lista de todos los ficheros en / y (separadamente) /usr antes de que aprendiera mejores formas de encontrar ficheros.

`passwd`
: para cambiar la contraseña del usuario (o la contraseña de `root`)

`man hier`
: página del manual sobre el sistema de ficheros UNIX®

Usa `find` para localizar filename en /usr o cualquiera de sus subdirectorios con

```
% find /usr -name "filename"
```

Puedes usar * como un comodín en `"filename"` (que debería ir entre comillas). Si le pides a `find` buscar en / en lugar de /usr buscará los ficheros en todos los sistemas de ficheros montados, incluidos el CDROM y la partición DOS.

Un libro excelente que explica comandos UNIX® y utilidades es Abrahams & Larson, Unix for the Impatient (2nd ed., Addison-Wesley, 1996). También hay mucha información sobre UNIX® en Internet.

7. Próximos pasos

Ahora ya deberías tener las herramientas que necesitas para moverte y editar ficheros de forma

que puedas poner todo en funcionamiento. Hay mucha información en el FreeBSD handbook (que está probablemente en tu disco duro) y en El sitio web de FreeBSD. Hay una gran variedad de paquetes y ports en el CDROM así como en el sitio web. El handbook te dice más sobre cómo usarlos (obtener el paquete si existe, con `pkg add packagename` donde *packagename* es el nombre del paquete). El CDROM tiene listas de paquetes y ports con breves descripciones en cdrom/packages/index, cdrom/packages/index.txt, y cdrom/ports/index, con descripcones completas en /cdrom/ports/*/*/pkg/DESCR, donde * representa subdirectorios de tipos de programas y nombres de programas respectivamente.

Si encuentras el manual demasiado sofisticado (con el comando `lndir` y el resto) acerca de instalar los ports desde el CDROM, esto es lo que normalmente funciona:

Localiza el port que deseas, por ejemplo, `kermit`. Habrá un directorio para él en el CDROM. Copia el subdirectorio a /usr/local (un buen lugar para el software que instales y que debería estar disponible para todos los usuarios) con:

```
# cp -R /cdrom/ports/comm/kermit /usr/local
```

Esto debería resultar en un subdirectorio /usr/local/kermit que tiene todos los ficheros que están en el subdirectorio `kermit` del CDROM.

A continuación, si aún no existe, crea el directorio /usr/ports/distfiles usando el comando `mkdir`. Ahora busca en el directorio /cdrom/ports/distfiles un archivo que tenga el nombre del port que quieres. Copia ese archivo a /usr/ports/distfiles; en las versiones más recientes, puedes omitir este paso, FreeBSD lo hará por ti. En el caso de `kermit`, no existe el distfile.

Después usa `cd` para ir al subdirectorio /usr/local/kermit que tiene el fichero Makefile. Teclea

```
# make all install
```

Durante este proceso el port se conectará por FTP para obtener cualquier fichero comprimido que necesite y que no encontró en el CDROM o en /usr/ports/distfiles. Si no tienes la red funcionando todavía y no había fichero para el port en /cdrom/ports/distfiles, tendrás que obtener el distfile utilizando otra máquina y copiarlo a /usr/ports/distfiles. Lee Makefile (con `cat` o `more` o `view`) para averiguar dónde ir (el sitio maestro) para obtener el fichero y saber cuál es su nombre. (¡Utiliza transferencias binarias!) Después vuelve a /usr/local/kermit, encuentra el directorio que contiene el fichero Makefile, y teclea `make all install`.

8. Tu entorno de trabajo

Tu shell es la parte más importante de tu entorno de trabajo. Ese shell interpreta los comandos que escribes en la línea de comandos y, por lo tanto, se comunica con el resto del sistema operativo. También puedes escribir shell scripts, que consisten en una serie de comandos que se ejecutarán sin intervención.

Con FreeBSD vienen dos shells instalados: `csh` y `sh`. `csh` es bueno para trabajar en línea de comando,

pero los scripts deberían escribirse en `sh` (o `bash`). Puedes averiguar qué shell tienes tecleando `echo $SHELL`.

El shell `csh` está bien, pero `tcsh` hace todo lo de `csh` y más. Te permite recordar comandos con las flechas de dirección y editarlos. Tiene auto completado de ficheros con el tabulador (`csh` utiliza `Esc`), y te permite cambiar al último directorio en el que estuviste con `cd -`. El prompt también es mucho más fácil de cambiar en `tcsh`. Hace la vida mucho más fácil.

Aquí tiene los 3 pasos necesarios para instalar una nueva shell:

1. Instala el shell como un port o paquete, tal como harías con cualquier otro port o paquete.
2. Utiliza `chsh` para cambiar tu shell a `tcsh` permanentemente, o teclea `tcsh` en el prompt para cambiar tu shell sin iniciar una nueva sesión.



Puede ser peligroso cambiar el shell de `root` a algo que no sea `sh` o `csh` en versiones antiguas de FreeBSD y muchas otras versiones de UNIX®; podrías no tener un shell que funcione cuando el sistema te pone en modo usuario único. El solución es usar `su -m` para convertirse en `root`, lo que te dará un `tcsh` como `root` porque el shell es parte del entorno. Puedes hacer que esto sea permanente añadiéndolo un alias a tu .tcshrc con:

```
alias su su -m
```

Cuando `tcsh` se inicia, leerá los ficheros /etc/csh.cshrc y /etc/csh.login, como hace `csh`. También leerá .login en tu directorio home así como .cshrc a menos que proporciones un .tcshrc. Esto lo puedes hacer simplemente copiando .cshrc a .tcshrc.

Ahora que tienes instalado `tcsh` puedes ajustar tu prompt. Puedes encontrar los detalles en la página de manual de `tcsh`, pero aquí tienes una línea para poner en tu .tcshrc que te dirá cuántos comandos has tecleado, qué hora es, y en qué directorio estás. También utiliza un `>` si eres un usuario ordinario y un `#` si eres `root`, pero tcsh lo hará de todos modos:

set prompt = "%h %t %~ %# "

Debería de ir en el mismo lugar que la línea del prompt actual, si existiera, o debajo de "if($?prompt) then" si no existiera. Comenta la línea antigua; siempre podrá volver a usar el método antiguo si lo prefieres. No olvides los espacios y las comillas. Puedes forzar la relectura del archivo .tcshrc escribiendo `source .tcshrc`.

Puedes obtener un listado de las otras variables de entorno que han sido configuradas ejecutando `env` en el prompt. El resultado mostrará tu editor predeterminado, paginador y tipo de terminal, entre muchas otras. Un comando útil si inicias sesión desde una ubicación remota y no puedes ejecutar un programa porque el terminal no es capaz de hacerlo es `setenv TERM vt100`.

9. Otros

Como `root`, puedes desmontar el CDROM con `/sbin/umount /cdrom`, sacarlo de la unidad, insertar otro, y montarlo con `/sbin/mount_cd9660 /dev/cd0a /cdrom` asumiendo que cd0a es el nombre de la unidad para tu CDROM. Las versiones más recientes de FreeBSD te permiten montar el CDROM con tan sólo `/sbin/mount /cdrom`.

Utilizar el sistema de archivos live-el segundo disco de los CDROM de FreeBSD- es útil si tienes un espacio limitado. El contenido del sistema de archivos live varía de una versión a otra. Puedes probar a jugar a los juegos que hay en el CDROM. Esto implica usar el comando `lndir`, que se instala junto al sistema de ventanas X (X Window System), para informar al resto de programas dónde encontrar los archivos necesarios, dado que se encuentran en /cdrom en lugar de /usr y sus subdirectorios, que es donde se espeara que estén. Lee `man lndir`.

10. Comentarios Bienvenidos

Si utilizas esta guía, me interesaría saber qué partes no han quedado del todo claras y qué echas en falta y piensas que debería incluirse, y si te fue útil . Gracias a Eugene W. Stark, profesor de ciencias de la computación en SUNY-Stony Brook y a John Fieber por sus útiles comentarios.

Annelise Anderson, andrsn@andrsn.stanford.edu