Departamento de
Automática y Computación
Automatika eta
Konputazio Saila

Campus de Arrosadía
Arrosadiko Campusa
31006 Pamplona - *Iruñea*
Tfno. 948 169113, Fax. 948 168924
Email: ayc@unavarra.es

Práctica1 - Introducción a OPNET y Linux

1- Objetivos

El objetivo principal que se persigue en esta práctica es que el alumno se familiarice con el entorno de trabajo en el que se desarrollarán las prácticas posteriores: Linux y el simulador de redes de comunicaciones orientado a eventos; OPNET(en entorno Windows).

En esta segunda parte se aprenderá el uso más básico de un sistema operativo Linux. Gran parte de los conceptos no son específicos de Linux sino que pueden aplicarse igualmente a cualquier sistema operativo de tipo UNIX.

2- LINUX

Interfaz gráfico

Consiguiendo una cuenta

Los sistemas UNIX son multiusuario, permitiendo que el mismo equipo sea usado por varios usuarios (incluso al mismo tiempo). Los usuarios pueden utilizar el ordenador tanto de forma local (estando sentados delante) como de forma remota (dándole órdenes a través de la red desde otro ordenador). Normalmente los propietarios de ordenadores UNIX no quieren que cualquiera pueda utilizar su ordenador así que es necesario que el sistema operativo UNIX almacene los datos de los usuarios que tienen derecho a usarlo. A ésto se le llama cuenta. La cuenta se identifica por un nombre o identificador de usuario y una contraseña que se supone que sólo es conocida por el usuario autorizado y que le permite probar su identidad.

Cada cuenta permite a su usuario utilizar una máquina UNIX con diferentes privilegios que otros usuarios, pudiendo acceder sólo a los ficheros que sean propiedad de ese usuario y utilizando las aplicaciones que se permitan a ese usuario. Hay una cuenta especial en las máquinas UNIX, es la cuenta del administrador del sistema. Esta cuenta se llama normalmente root y tiene permisos para hacer cualquier cosa en el sistema.

Al proceso de identificarse en el sistema mediante el nombre de cuenta y contraseña antes de utilizar el sistema lo llamaremos hacer login.

El primer paso, por tanto, para usar un sistema UNIX es conseguir una cuenta. A cada alumno se le ha asignado una cuenta que le permite utilizar cualquiera de los PCs de propósito general del laboratorio. Esta cuenta se llamará arss<número> y deberá apuntarse para obtenerla, el profesor le comunicará la contraseña en clase. Una vez conozca el nombre de la cuenta y la contraseña debe apropiarse de la cuenta que tiene asignada. Para ello debe cambiar la contraseña a una que sólo conozca cada uno.

Entre en su sistema UNIX para PC, linux, (haga login) introduciendo el nombre de la cuenta y la contraseña en el interfaz gráfico. El ordenador arrancará el sistema gráfico con el escritorio para su

Campus de Arrosadía
Arrosadiko Campusa
31006 Pamplona - *Iruñea*
Tfno. 948 169113, Fax. 948 168924
Email: ayc@unavarra.es

cuenta. Lo primero de todo busque un navegador Web. Introduzca la dirección: http://www.tlm.unavarra.es y busque la página de esta asignatura. A partir de aquí podrá consultar los guiones de prácticas en la Web mientras los hace, siendo recomendable tener el guión impreso, así como la documentación de los armarios.

Antes de probar a utilizar más aplicaciones debe cambiar la contraseña a su contraseña privada. Para ello siga las instrucciones:

- Lance una aplicación de terminal (también llamada de consola) que le permita darle comandos de texto al sistema operativo. Normalmente estará en el menú desplegable que puede sacar con el botón derecho en el escritorio. Elija "Open Terminal". Si no, busque en el menú de "red-hat" (abajo a la izquierda) "System Tools" > "Open Terminal".
- Elija una contraseña nueva que proteja su cuenta. Una buena contraseña debe ser suficientemente larga y ser difícil de averiguar. Los consejos típicos son que no se usen palabras que estén en el diccionario y para eso se recomienda normalmente usar letras mayúsculas y minúsculas mezcladas con números y símbolos. También es bueno que sea fácil de recordar para su dueño para evitar escribirla, si no está escrita nadie la puede ver. Usen el sentido común y piensen una contraseña para su cuenta.
- Utilice el comando `yppasswd`, para cambiar la contraseña de su cuenta en la red de Telemática. Normalmente, el comando para cambiar la contraseña en una máquina UNIX es passwd. Sin embargo en un sistema como el del laboratorio, las cuentas no pertenecen a un solo ordenador sino que funcionan en varios ordenadores porque están almacenadas en un ordenador central. En estos casos puede haber otros comandos para cambiar la contraseña. En el caso del laboratorio se hace con el comando yppasswd.

```
$ yppasswd
Changing NIS account information for arss on bender.net.tlm.unavarra.es.
Please enter old password:     [metemos aquí la contraseña actual ]
Changing NIS password for arss on bender.net.tlm.unavarra.es.
Please enter new password:     [metemos aqui la contraseña nueva  ]
Please retype new password:    [metemos otra vez la contraseña nueva para
                               verificar]

The NIS password has been changed on bender.net.tlm.unavarra.es.
$
```

Observe cómo el comando nos informa de que la contraseña ha sido cambiada correctamente. En caso de que no haya sido así, descubra por qué y cámbiela correctamente. Tenga en cuenta que puede fallar si no pone bien la contraseña actual o si no escribe la misma nueva contraseña dos veces. También es posible que el comando se queje si intenta poner una contraseña demasiado fácil.

- Pruebe a salir de la cuenta y vuelva a entrar para comprobar que la contraseña original ya no funciona y la nueva contraseña funciona. Asegúrese que su cuenta tiene una nueva contraseña antes de continuar.

Nota: *Tenga en cuenta que tener los ordenadores con contraseñas conocidas es un problema grave de seguridad, así que las cuentas que sigan teniendo la misma contraseña se considerará que no tienen propietario y serán eliminadas.*

Departamento de
Automática y Computación
Automatika eta
Konputazio Saila

Campus de Arrosadía
Arrosadiko Campusa
31006 Pamplona - *Iruñea*
Tfno. 948 169113, Fax. 948 168924
Email: ayc@unavarra.es

Familiarícese con el escritorio

Como ha visto, sólo tiene que poner su cuenta y su contraseña y entrará al sistema en modo gráfico. En breves instantes tendrá el escritorio de Linux. El sistema gráfico de UNIX se llama X (o X-Window o X11) que es el motor que dibuja en pantalla. Sobre X puede haber un administrador de ventanas o de escritorio que es el programa que le dice a X como dibujar las ventanas y los demás elementos de interfaz de usuario. Al ser Linux de código abierto y muy modular se han escrito varios de estos administradores del interfaz. Puede elegir entre varios (aunque conforme van evolucionando cada vez se parecen más) los nombres de los principales entornos gráficos son Gnome y KDE. Buscando un poco podrá elegir entre esos y algún otro. Supondremos que ya tiene experiencia en moverse por un escritorio gráfico con un ratón, así que experimente por su cuenta. Pero debe buscar y aprender a hacer al menos las siguientes cosas:

1. Ya ha probado como lanzar un terminal. Un terminal es una aplicación que le permite abrir una sesión de comandos. El terminal lanza un programa llamado shell que le permite escribir comandos o lanzar programas. Hay varias *shells* diferentes disponibles normalmente en los sistemas UNIX que el usuario puede elegir. En el Linux que está usando si el usuario no elige lo contrario se utiliza la *shell* llamada bash. Otras *shells* típicos son csh, tcsh, ksh y sh. UNIX es un sistema operativo multitarea que puede tener varios programas funcionando al mismo tiempo. A cada programa que está corriendo le llamamos proceso. Así pues, cuando escribe algo en la shell y pulsa ENTER, se lanza un nuevo proceso con el programa que ha introducido. La shell se queda esperando a que el proceso termine y vuelve a preguntarle por otro proceso. Aunque vaya a usar las X en la mayoría de las prácticas necesitará seguir escribiendo comandos.
2. Localice, si no lo ha hecho ya, un navegador de Web.
3. Busque un editor de texto, los clásicos son vi o emacs, quizás le parezca que tienen un aspecto anticuado, pero siguen siendo útiles porque pueden usarse en un interfaz de texto sin gráficos. Puede probarlos lanzándolos desde un terminal:

```
$ vi nombredefichero
$ emacs nombredefichero
```

   Si prefiere algo más fácil, hay cientos de editores donde probar. Busque por ejemplo gedit (del Gnome), kedit (del KDE), aunque por aquí preferimos nedit. Elija el que más le guste.

4. En cualquier caso sepa que puede lanzar los editores desde el terminal, lo que es útil si está escribiendo comandos y quiere abrir un fichero desde el terminal, sólo tiene que escribir el nombre del editor seguido del fichero:

```
$ nedit nombredefichero
```

   El editor es sin más un programa que puede abrir ventanas. Con eso puede editar el fichero de forma más cómoda. Sin embargo fíjese que mientras el programa nedit está funcionando la shell se ha quedado esperándole, con lo que no puede hacer mas comandos en esa shell. Puede usar el comando con un **&** detrás para que la shell no espere a que acabe el programa.

```
$ nedit nombredefichero &
```

5. Finalmente puede buscar un explorador gráfico del sistema de ficheros y buscar su directorio Home (en UNIX se llama así al directorio asignado a cada usuario que es el único en el que debería poder guardar cosas). Puede usarlo también para copiar y mover ficheros. En cualquier caso fíjese que tiene su directorio Home y que dentro de él tendrá un directorio llamado Desktop que corresponde a lo que tiene en el escritorio.

Interfaz modo texto

Observe que con la combinación de teclas `Control+Alt+F1` puede elegir un login en modo texto (con `Control+Alt+F7` se vuelve al interfaz gráfico).

Entre utilizando su cuenta y contraseña en el login de modo texto. Una vez identificado mediante usuario y contraseña, el sistema ha lanzado un proceso que recibe las órdenes que teclea y las interpreta, lanza los programas que sean necesarios para cumplirlas y muestra los resultados en pantalla. Este programa se llama genéricamente *shell* (ya comentado y que en el modo gráfico lanzamos al abrir un terminal).

Empezaremos por un ejemplo sencillo. Pruebe con un comando (programa), escriba:

```
$ date  <ENTER>
```

(A partir de ahora las líneas como la anterior que empiecen por $ se supone que son comandos para la shell) Obtendrá como resultado algo como:

```
$ date
Tue Feb 3 19:04:21 CET 2009
$
```

La shell ha lanzado el programa date y una vez terminado vuelve a aceptar órdenes. Pruebe estos otros comandos y observe qué hacen:

```
$ ls
$ clear
$ exit
$ uname
$ hostname
$ who
$ cal
```

Normalmente el uso de los comandos no es tan simple como escribir su nombre. Los comandos tienen argumentos y opciones que se escriben a continuación de su nombre.

Por ejemplo el comando `cal` que ha probado antes, puede usarse de varias formas:

```
$ cal
$ cal 6 2005
$ cal 2006
```

Departamento de
Automática y Computación
Automatika eta
Konputazio Saila

Campus de Arrosadía
Arrosadiko Campusa
31006 Pamplona - *Iruñea*
Tfno. 948 169113, Fax. 948 168924
Email: ayc@unavarra.es

Para no tener que recordar todas las opciones existe un sistema de ayuda en UNIX proporcionado por el comando `man`.

```
$ man [ seccion ] comando
$ man -k palabraclave
```

Con la primera forma pedimos ayuda sobre un comando y `man` le mostrará una o bastantes páginas de ayuda sobre ese comando, incluyendo en las primeras líneas un resumen de la sintaxis del comando y el significado de cada una de las opciones. Este manual en línea se estructura en secciones y se puede especificar en qué sección queremos mirar. La sección 1 es para los comandos, las secciones mayores son para las funciones del sistema, información para programación y ficheros de configuración. En general todo lo que busque en estas prácticas estará en la sección 1.

Con la opción -k podemos decirle que nos busque comandos que en su explicación contenga dicha palabra clave. Esto vale si estamos buscando un comando que haga algo concreto pero no nos acordamos cómo se llama el comando.

Pruebe a buscar en el `man` información de otros comandos:

```
$ man cal
$ man date
$ man man
```

Caracteres especiales

Hay algunas combinaciones de teclas que son interpretadas como acciones por un terminal UNIX. Estas combinaciones se consiguen pulsando la tecla `Control` a la vez que otra letra y se suelen escribir como **^[letra]**. Ciertas combinaciones son de especial interés y debe conocerlas:

^C interrumpe un proceso que deja de ejecutarse y devuelve el control a la shell.

^Z detiene un proceso que se queda suspendido y devuelve momentáneamente el control a la shell pero el proceso puede volver a reanudarse.

^D fin de entrada, le dice a un proceso que está esperando datos del usuario que ya hemos acabado de introducir los datos.

^S detiene la salida de texto por pantalla manteniendo fijo el contenido.

^Q reanuda la salida de texto.

Y éstos que no son tan útiles pero por comentarlos:

^H borra el último carácter (lo mismo que la tecla backspace <-)

^W borra una palabra

^U borra toda la línea

Observe el funcionamiento de **^C**. Para eso necesita un proceso que no acabe y le de tiempo a pulsar **^C**. Por ejemplo el comando "ping". Mire en el "man" cómo se usa.

Observe las diferencias de **^C** y **^Z**. En la siguiente práctica veremos más sobre suspender y reanudar procesos, pero de momento puede reanudar el último proceso suspendido con el comando `fg`.

Observe el funcionamiento de **^S** y **^Q**. Para ello pulse **^S**. Intente escribir en pantalla. Pulse <ENTER>
ls<ENTER>.
Pulse ahora **^Q**. ¿Qué está pasando?

Recuerde ésto si en alguna ocasión le parece que su ordenador está colgado.

Saliendo de la cuenta (logout)

Al terminar de usar una máquina UNIX debe indicar a la shell que ya no quiere hacer más operaciones, para que termine y vuelva a pedir login al siguiente usuario. Para ello teclee `exit` y pulse <ENTER>. Ésto es muy importante por motivos de seguridad. Si olvida cerrar su sesión cualquiera delante de ese ordenador puede acceder a su cuenta, lo que implica que puede borrarle todos sus ficheros o usar todos los privilegios de su cuenta. Acuérdese siempre de dejar un ordenador UNIX en la pantalla de login igual que cuando entró.

Por otra parte en general no debe apagar las máquinas UNIX. No apague los ordenadores del Laboratorio de Telemática, se considera de mala educación hacerlo. Recuerde que puede haber varios usuarios y que otros usuarios pueden estar utilizando ese ordenador de forma remota al mismo tiempo y no les hará gracia que se les apague de repente.

Checkpoint 1.1: Muestre al profesor de prácticas que ha cambiado la contraseña de su cuenta.

Nota: Realice la práctica desde el siguiente apartado y hasta el final de la misma en uno de los PCs A, B ó C. Recuerde que en estos PCs debe emplear el nombre de cuenta ***`arss`*** *y la contraseña* ***`telemat`***.

El superusuario

En las prácticas deberá configurar el sistema operativo Linux para utilizar la red. Sin embargo algunas de las operaciones que nos permiten cambiar parámetros de red requieren tradicionalmente en UNIX privilegios de superusuario. Los ordenadores disponibles para prácticas tendrán el sistema operativo preparado para que pueda realizar las operaciones necesarias de configuración de red sin necesidad de utilizar la cuenta de root. Aun así, es interesante que sepa como funciona el acceso a la cuenta del administrador.

En los sistemas UNIX suele existir una cuenta de administrador del sistema más conocida como root. Se puede entrar en el sistema como root conociendo la contraseña de dicha cuenta. Pero no sólo se puede utilizar un usuario haciendo login con dicha cuenta, también se puede usar el comando su que permite cambiar de usuario (en realidad sacar una shell propiedad de otro usuario). Pruebe ésto:

```
$ su lc
```

E introduzca la contraseña de `lc` *("telemat").* Obtendrá una nueva shell del usuario `lc`. La shell mantiene el mismo directorio actual que la shell inicial. Si lo prefiere puede usar la opción -

```
$ su – lc
```

En este caso su nueva shell será la del usuario `lc` y además estará en el *home* de `lc`. La nueva shell será igual que si `lc` hubiera hecho login.

El comando `su` también nos permite cambiar a la cuenta de root. De hecho si no le decimos el usuario de destino entenderá que queremos ser administrador.

```
$ su
    Conviértete en administrador
$ su -
    Conviértete en administrador como si root hubiera hecho login
```

Para usar el comando `su` sigue necesitando conocer la contraseña de administrador. Pero hay otro comando; `sudo`, que le permite realizar tareas de administrador sin conocer la contraseña. El comando `sudo` permite que se especifiquen permisos para realizar ciertas tareas a ciertos usuarios y es el que se usará en las prácticas para permitir que los alumnos tengan permisos de `root` para ciertas tareas.

Para usar el comando `sudo` debe escribir `sudo` y a continuación el comando que quiere realizar, que normalmente sólo podría usar el administrador del sistema. Por ejemplo para realizar el comando `ifconfig eth0 down` (que deshabilita el interfaz de red `eth0`) deberá realizar:

```
$ sudo ifconfig eth0 down
Passwd:
```

El comando `sudo` puede pedir la contraseña, pero tiene que utilizar su contraseña y no la del administrador. El comando no la pide para verificar que es el administrador (como hace `su`) sino para verificar que es el usuario al que se le han dado permisos para utilizar ese comando.

Introduciendo su contraseña se ejecutará el comando. `sudo` no le pedirá la contraseña en todas las ocasiones, normalmente se acuerda de que es un usuario de fiar durante algún tiempo. De todas formas también se puede configurar que para algunos comandos no pida ni siquiera contraseña y así es como está configurado en los ordenadores de prácticas.

Puede averiguar el listado de comandos de los que dispone para ejecutar como `root` mediante `sudo`. Consulte para ello el manual correspondiente, como ya hiciera en apartados anteriores.

Checkpoint 1.2: Muestre al profesor de prácticas la lista de comandos obtenidos y que es capaz de ejecutar como `root` las aplicaciones `ethereal` y `tcpdump`.

Departamento de
Automática y Computación
Automatika eta
Konputazio Saila

Campus de Arrosadía
Arrosadiko Campusa
31006 Pamplona - *Iruñea*
Tfno. 948 169113, Fax. 948 168924
Email: ayc@unavarra.es

Cambiando el PATH

Hasta ahora cuando ha utilizado un comando, la shell se ha encargado de encontrar el programa adecuado en el disco y lanzar el programa correspondiente. Esto es así porque los programas que hemos usado son programas comunes y estaban almacenados en los directorios en que tipicamente se guardan los comandos de UNIX (/bin,/usr/bin)

Puede probar el comando `which` que le indica donde ha encontrado la shell un comando concreto.

```
$ which date
$ which cat
```

Podemos indicar a la shell la localización del programa que queremos ejecutar indicando el camino al comando en lugar del comando:

```
$ /bin/cat /etc/services
```

Tiene el mismo efecto que:

```
$ cat /etc/services
```

Con la única diferencia que la shell no busca el comando en el disco duro sino que va directamente a /bin a buscarlo

Pero aún en el caso en que la shell deba buscar el fichero del comando, no hace una búsqueda por todo el disco, sino que simplemente mira en una lista de directorios en los que comúnmente están los comandos. Esta lista se almacena en una variable de entorno del shell que se denomina `PATH`.

Puede ver el valor de esa variable con el siguiente comando. (La sintaxis para acceder a las variables en la shell es poner un símbolo $ delante del nombre de la variable que será sustituida por su valor)

```
$ echo $PATH
/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/sbin:/opt3/staff/usuario/bin
```

Observe que el contenido de la variable es una serie de directorios separados por ":". La shell busca comandos sólo en esos directorios y lo hace en el orden en que aparecen en la variable. Modificando la variable conseguirá que busque en otros directorios. Por ejemplo es probable que en las prácticas con la red, quiera acceder a comandos que estén en el directorio /sbin. Puede añadir el directorio /sbin a la variable PATH como se indica a continuación (tenga en cuenta que las variables se manejan de forma diferente en las diferentes tipos de shell, las instrucciones que siguen sólo son aplicables a las shells de tipo `bash`).

Así se da valor a una variable de la shell:

```
$ A=1
```

Veamos si funciona:

```
$ echo "A=$A"
```

Así se da valor a una variable que será heredada por los procesos hijos:

```
$ export A=1
```

Así podemos modificar el PATH:

```
$ export PATH=$PATH:/sbin
```

```
$ echo "PATH=$PATH"
```

Compruebe que una nueva shell lanzada desde ésta recibe la variable PATH modificada.

Compruebe también que la variable sólo se propaga a los procesos descendientes de la shell en la que la hemos cambiado. Si, por ejemplo, abre un terminal nuevo, no tendrá la variable.

Para modificar la variable PATH de forma que afecte a cualquier nueva shell que abra deberá modificar el fichero de arranque de la shell. Cuando se lanza `bash` lee una serie de ficheros entre los que se encuentra el fichero `.bashrc` en su directorio *home* y ejecuta los comandos de esos ficheros. El fichero `.bashrc` es el fichero apropiado para poner modificaciones a la variable PATH. Puede editarlo y colocar ahí la línea que modifique la variable. Compruebe que todas las shells que abre ahora tienen la variable modificada.

Checkpoint 1.3: Muestre al profesor de prácticas cómo tendría que modificar su fichero .bashrc para poder ejecutar el programa “ethereal” sin necesidad de indicar la ruta de acceso al mismo.

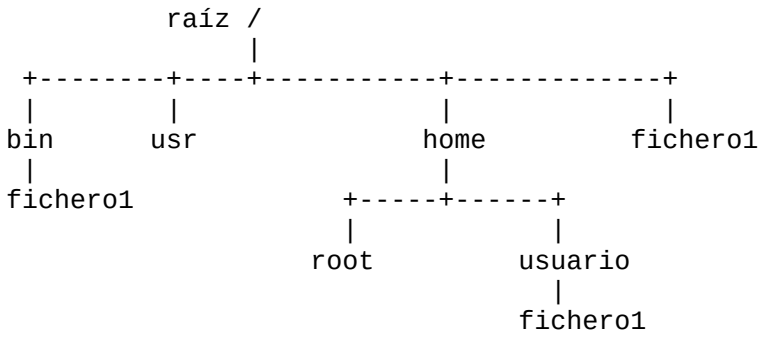
Justifique el hecho de que el sistema le pida contraseña al realizar la ordern `sudo ethereal` a pesar de haber modificado correctamente su `.bashrc`.

Uso del sistema de ficheros

Una de las funciones básicas del sistema operativo UNIX (y de los otros) es permitirnos trabajar con los datos almacenados en los dispositivos de almacenamiento mediante sistemas de ficheros. Los datos en un dispositivo de almacenamiento (normalmente el disco duro) se nos presentan como una jerarquía de directorios dentro de los cuales están los datos agrupados en ficheros.

En UNIX todos los ficheros se distribuyen en una jerarquía única en forma de árbol. El directorio que contiene a todos es el directorio "raíz" (o "root") representado por /

Dentro de / puede haber ficheros u otros directorios que a su vez pueden contener mas ficheros y directorios. Como en el ejemplo:



Gracias a la estructura de directorios podemos tener ficheros con el mismo nombre como en el ejemplo, siempre que no estén en el mismo directorio. Para identificar unívocamente un fichero en un sistema de ficheros de UNIX empleamos el *camino absoluto* formado por la lista de todos los directorios que contienen al fichero empezando por / (y usando / como separador). En el ejemplo

Campus de Arrosadía
Arrosadiko Campusa
31006 Pamplona - *Iruñea*
Tfno. 948 169113, Fax. 948 168924
Email: ayc@unavarra.es

anterior existen 3 ficheros con el nombre fichero1 identificados completamente por sus caminos absolutos:

```
/fichero1
/bin/fichero1
/home/usuario/fichero1
```

Otra forma de referirse a un fichero en UNIX es a través de un *camino relativo*. Cada proceso (programa corriendo) en UNIX está asociado a un directorio que es su *directorio de trabajo*. Un *camino relativo* expresa la posición de un fichero en relación con el *directorio de trabajo*. Por ejemplo, si un proceso que este corriendo con *directorio de trabajo* /home/usuario se refiere al fichero fichero1 en realidad se está refiriendo a /home/usuario/fichero1. Un proceso que esté corriendo en /home puede referirse a ese mismo fichero como usuario/fichero1.

Para especificar caminos relativos se aceptan dos directorios especiales dentro de cada directorio:

```
.       se refiere al directorio actual
..      se refiere al directorio padre, es decir baja un nivel
```

Es decir que un proceso que esté corriendo en /home puede referirse al fichero que está en /fichero1 con el camino relativo ../fichero1 y un proceso que esté corriendo en /home/usuario puede usar ../../bin/fichero1 para referirse a /bin/fichero1

En UNIX dispone de diversos comandos para moverse y manejar el sistema de ficheros. El objetivo de lo que queda de práctica es que se acostumbre al uso de los comandos más básicos.

El primero es el comando `pwd` (de *print working directory*) le muestra el *directorio de trabajo* actual. Pruebe el comando:

```
$ pwd
```

Es el directorio de trabajo actual... ¿De qué proceso?

Observe que no está en el directorio /. Sino que está en un directorio propio de su cuenta. Este directorio suele ser `/home/nombredelacuenta` (PCs A, B ó C), o bien `/opt3/arss/nombredesucuenta` en el resto de equipos del laboratorio. Al conseguir una cuenta en UNIX ha conseguido también un espacio para sus propios ficheros. De hecho como UNIX es un sistema muy orientado a tener usuarios diferentes no tendrá permiso para hacer ficheros y directorios fuera de su directorio. Al directorio que obtiene por tener una cuenta se le suele llamar en la jerga UNIX directorio *home*. Así que cuando en estas prácticas nos refiramos a su *home* queremos decir `/home/nombredelacuenta` ó `/opt3/arss/nombredesucuenta`, en los equipos respectivos anteriormente comentados, y no a `/home`.

El comando `cd` nos permite cambiar el directorio actual, es decir nos permite movernos por el sistema de ficheros. Pruebe a moverse por los directorios de su ordenador. Siga este camino haciendo `pwd` a cada paso para observar si está en el directorio que cree:

```
$ cd /
$ cd usr
$ cd local
```

Departamento de
Automática y Computación
Automatika eta
Konputazio Saila

Campus de Arrosadía
Arrosadiko Campusa
31006 Pamplona - *Iruñea*
Tfno. 948 169113, Fax. 948 168924
Email: ayc@unavarra.es

```
$ cd ../bin
$ cd ../..
$ cd ..
$ cd ..
$ cd
$ cd ../../../usr/local/../bin
$ cd
$ cd /etc
$ cd /usr/./bin
```

¿Qué hace el comando cd sin argumentos?

```
$ cd
```

Vamos ahora a crear ficheros y directorios. Vaya a su directorio *home*. Lance el siguiente comando:

```
$ cat >mifichero
```

El ordenador se quedará esperando a que escriba algo, escriba texto, varias líneas. ¿Cómo puede decirle al programa que ya no quiere escribir más? Recuerde lo que ha leído antes, sólo tiene que pulsar:

```
^D
$   [ de nuevo la shell tiene el control ]
```

Ya está creado el fichero con nombre mifichero. Para comprobarlo tendrá que aprender a listar los ficheros que hay en un directorio. Use el comando `ls`

```
$ ls
mifichero   [ahí está]
$
```

El comando `ls` permite listar el contenido de un directorio y acepta un gran número de opciones para modificar el formato de presentación. Consulte la información del comando `ls` con el comando `man` y busque como obtener los opciones más útiles:

- Mostrar los ficheros ocultos. Qué tipo de ficheros se ocultan?
- Mostrar información asociada a cada fichero (formato largo)

En el formato largo debe obtener la lista de ficheros con este aspecto:

```
drwxr-xr-x   3 usuario staff     4096 Dec 16 08:14 Desktop
-rw-r--r--   1 usuario staff 41685513 Sep 27 13:57 usuario.tgz
-rw-r--r--   1 usuario staff  1000000 Nov 12 20:00 oo
drwxr-xr-x   2 usuario staff     4096 Oct  6 13:07 prueba_c
-rwxr-xr-x   1 usuario staff       94 Nov 25 11:56 vnc_to_usuario.bash
drwx------  10 usuario staff     4096 Oct  6 11:26 W2000
```

Cada fichero aparece como una línea con varios campos. El formato de cada uno de estos campos es el siguiente:

Departamento de
Automática y Computación
Automatika eta
Konputazio Saila

Campus de Arrosadía
Arrosadiko Campusa
31006 Pamplona - *Iruñea*
Tfno. 948 169113, Fax. 948 168924
Email: ayc@unavarra.es

```
drwxr-xr-x   3 usuario staff     4096 Dec 16 08:14 Desktop
```

`[d]`	La primera letra indica el tipo de fichero: - significa un fichero. d significa un directorio.
`[ rwxr-xr-x ]`	Indica los permisos del fichero, quién puede leer(**r**ead), escribir(**w**rite) y ejecutar(e**x**ecute) el fichero. `rwx` permisos del propietario `r-x` permisos del grupo `r-x` permisos de los demás
`[ 3 ]`	Éste se deja para asignaturas avanzadas.
`[ usuario ]`	El nombre de usuario propietario del fichero
`[ staff ]`	El nombre del grupo de usuarios propietario del fichero.
`[ 4096 ]`	El tamaño del fichero. En el caso de un directorio no es lo que ocupa el directorio sino el tamaño necesario para almacenar la información del directorio.
`[ Dec 16 08:14 ]`	La fecha de la última modificación se puede cambiar para que sea el último acceso.
`[ Desktop ]`	El nombre del fichero o directorio.

Las operaciones son ligeramente diferentes según se trate de ficheros o directorios. Puede verlas en la siguiente tabla:

Operación	Sobre ficheros	Sobre directorios
r leer	El usuario puede leer el contenido del fichero	El usuario puede leer el contenido del directorio, esto es, ver la lista de ficheros que hay dentro
w escribir	El usuario puede modificar el contenido del fichero	El usuario puede modificar el conetnido del directorio, es decir añadir o quitar ficheros
x ejecutar	El usuario puede ejecutar el fichero. Sólo tiene sentido si en el fichero hay un programa	El usuario puede acceder a los ficheros y subdirectorios dentro del directorio (puede hacer cd a ese directorio)

Observe, con `ls`, cuántos bytes ocupa el fichero que ha creado. También puede usar el comando `wc`, consulte en el manual cómo hacerlo.

Para examinar el contenido de un fichero podemos usar varios comandos. El más simple es cat. Haga

```
$ cat mifichero
[ aquí aparecerá el contenido ]
$
```

`cat` es el comando básico para leer datos de cualquier sitio y enviarlos a otro. Antes hemos usado cat para leer datos del teclado y enviarlos a un fichero. Pero si le indicamos un fichero y no le decimos que lo guarde en un fichero con >mifichero, nos mostrará su salida por pantalla. Pero cat está pensado para copiar datos de un sitio a otro y no para verlos. Qué pasa si el fichero es

Departamento de
Automática y Computación
Automatika eta
Konputazio Saila

Campus de Arrosadía
Arrosadiko Campusa
31006 Pamplona - *Iruñea*
Tfno. 948 169113, Fax. 948 168924
Email: ayc@unavarra.es

tan grande que no cabe en una pantalla? Pruebe a ver el contenido del fichero /etc/services concat y lo comprobará

Hay otros comandos más orientados a ver el contenido de un fichero. Pruebe las diferencias entre estos

```
$ cat mifichero
$ more mifichero
$ head mifichero
$ tail mifichero
```

¿Hay diferencias? Pruebe con /etc/services

```
$ cat /etc/services
$ more /etc/services
$ head /etc/services
$ tail /etc/services
```

Consulte el manual de `head` y `tail` y pruebe a cambiar el número de líneas que muestran.

Checkpoint 1.4: Demuestre al profesor de prácticas que ha encontrado cómo contar las líneas y letras que tiene su fichero. Muéstrele también las diferencias entre cat, head, tail y more.