

PRACTICO 1
Introducción al uso del sistema operativo Linux.

Objetivos :

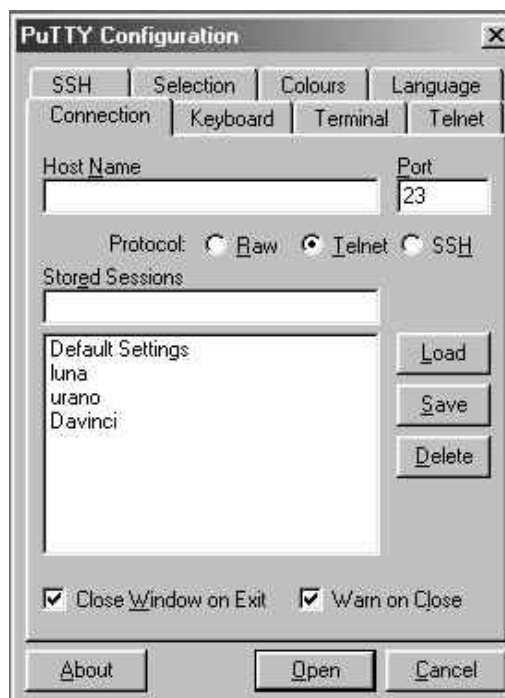
- Aprender a conectarse con la maquina desde una terminal remota e iniciar una sesión.
- Reconocer las características, elementos y funciones del sistema operativo Linux.
- Explorar las opciones disponibles para la visualización y edición de archivos.
- Acceder al programa Polyrate v 8.0 y familiarizarse con su estructura.

1. Conexión con la máquina.



La utilidad que se empleará para conectarse con el servidor remoto se llama Putty y se puede acceder a ella desde el icono que se encuentra en el Escritorio.

A continuación se muestra la ventana principal del programa que se obtienen como consecuencia de ejecutar Putty:



- En **Host Name** se escribirá la dirección del servidor remoto al cual nos vamos a conectar. En este caso es **luna.fcien.edu.uy** . Es importante recordar que debe marcarse la opción SSH (Secure Shell) para establecer una conexión segura (esto significa que la información viajará encriptada por la red). Antiguamente se solía usar el protocolo de comunicación Telnet pero este quedó en desuso por ser poco seguro.

Para no tener que volver a escribir toda la información necesaria para iniciar una sesión en un servidor remoto es posible guardar las características de una sesión. Para ello debe realizarse el siguiente procedimiento:

- Se escribe en **Host Name** la dirección de la máquina a la que se quiere conectar
- Se marca la opción **SSH**
- Se escribe en **Stored Sessions** un nombre con el que se identificará la sesión.
- Se marca **Save**

Una vez realizado este procedimiento, es posible iniciar la conexión remota haciendo doble clic sobre la sesión guardada.

2. Iniciar la sesión.

Luego de ejecutar Putty aparece la siguiente ventana llamada comúnmente terminal:



A continuación deberá autenticar su usuario frente al servidor. Para ello deberá escribir su nombre de usuario (junto a **Login as**) y su contraseña (junto a **Password**).

- Abrir una sesión en el servidor luna.fcien.edu.uy con el usuario y la contraseña que le haya proporcionado el instructor.

3. Familiarizándose con el sistema operativo Linux.

3.1. Introducción

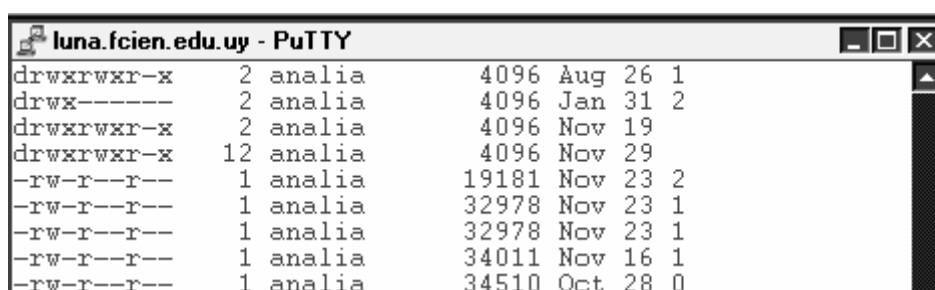
- Linux es un sistema operativo tipo Unix que tiene la característica fundamental de ser multiusuario (varios usuarios pueden trabajar a la vez) y multitarea real (permite llevar a cabo dos o más procesos al mismo tiempo). Windows es un sistema monousuario y multitarea por desalojo. Esto último significa que dos procesos pueden llevarse a cabo simultáneamente alternando de forma muy rápido entre ambos (no es multitarea real). Linux fue desarrollado por Linus Torwalds y ahora es mantenido fundamentalmente por la comunidad informática.
- Linux es un Sistema Operativo de código libre (open source), es decir que los códigos fuente de sus programas son de dominio público. En los comienzos Linux era un sistema adecuado solo para personas con muchos conocimientos técnicos, sin embargo actualmente empieza a ser visto como una alternativa a los Sistemas Operativos de Microsoft.
- Unix se originó en los laboratorios de Bell AT&T a comienzos de 1970 y MS-DOS tomó muchas de las ideas de Unix. Algunas de ellas son por ejemplo, el sistema de directorios en árbol y la estructura de varios comandos. Tradicionalmente los Sistemas Operativos Unix se ha caracterizado por ser poco intuitivos para el usuario ya que solo podían ser operados en modo texto (línea de comando). Esto está cambiando rápidamente por que Linux está ofreciendo cada vez más entornos gráficos más intuitivos para su utilización, que acercan al usuario doméstico. Ejemplo de esto último son los entornos gráficos KDE o Gnome.
- Una diferencia importante entre los Sistemas Operativos tipo Unix y los de Microsoft es que en Unix no existen las unidades lógicas, como ser c: , a: , d: . En Linux/Unix deberíamos hablar de sistemas de ficheros en lugar de unidades lógicas.
- Existen distintas distribuciones comerciales de Linux, como ser Slackware, Caldera, Debian y RedHat. Cada distribución tiene sus propias características que la hacen más adecuada para ciertos usos. Las distribuciones de Debian están optimizadas para actuar como servidores de red brindando servicios de correo electrónico, páginas Web, etc. RedHat en cambio es preferido como estación de trabajo.
- Nosotros trabajamos con RedHat versión 7.2. En el centro de cálculo de la Facultad de Ciencias se trabaja con Debian. RedHat es la distribución más ampliamente difundida,

Ofrece un entorno amigable que facilita la instalación. Debian resulta más adecuada para usuarios con conocimientos previos ya que es poco amigable.

- Existen también varios tipos de entornos o shells. Una shell es un intérprete de comandos y de lenguaje de programación. Por ejemplo encontramos C-Shell (csh) que emplea el lenguaje C++ y es útil para programadores. Otro shell muy usado es el Bourne-Again Shell (bash) que es más nueva e integra los mejores elementos del csh. Todos nuestros usuarios emplean bash como entorno. El uso de uno u otro shell tiene implicancias en como se definen y exportan las variables de entorno (una variable de entorno es por ejemplo la ruta a un determinado programa).

3.2. Comandos

- **Comando ls** : (list) Nos muestra el contenido del directorio sobre el cual estamos posicionados. Los directorios se ven en azul.
- **Comando ll** : (long list) Muestra todos los archivos contenidos en el directorio junto con información sobre dichos archivos.



```
luna.fcien.edu.uy - PuTTY
drwxrwxr-x  2 analia  4096 Aug 26 1
drwx----- 2 analia  4096 Jan 31 2
drwxrwxr-x  2 analia  4096 Nov 19
drwxrwxr-x 12 analia  4096 Nov 29
-rw-r--r--  1 analia 19181 Nov 23 2
-rw-r--r--  1 analia 32978 Nov 23 1
-rw-r--r--  1 analia 32978 Nov 23 1
-rw-r--r--  1 analia 34011 Nov 16 1
-rw-r--r--  1 analia 34510 Oct 28 0
```

Como ya se explicó anteriormente, Linux es un sistema multiusuario. Cuando un usuario va a usar Linux lo primero que necesita hacer es identificarse para ser autorizado a abrir una sesión de trabajo. También es multitarea, y por ello en un mismo instante varios procesos pueden estar funcionando y cada uno de ellos puede pertenecer a usuarios distintos. La información que se guarda en el disco duro también puede pertenecer a distintos usuarios. Para evitar que todo ello provoque conflictos es que hay cosas que están permitidas a ciertos usuarios y a otros no.

Cuando se usa el comando ll, sobre el lado izquierdo de la ventana aparecen una serie de letras a las que llamaremos permisos.

| | |
|--------------|-----------------------|
| drwxr -xr -x | Lectura = read: r |
| | Escritura = write : w |
| | Ejecución : x |

Se puede observar que en una misma línea de permisos, algunos permisos se repiten y otros están ausentes. Los permisos pueden dividirse en tres clases:

- La primera secuencia rwx corresponde al dueño de los archivos e indica si el usuario puede leer, escribir y ejecutar archivos.
- Los usuarios pueden estar organizados en diferentes grupos a fin de poder manejar permisos a nivel de grupo. Esto se hace para simplificar la administración del sistema. La segunda secuencia rwx pertenece al grupo (por ej. CTQT) e indica si todos los del grupo están autorizados a acceder a los archivos del usuario y en que forma.
- La tercera secuencia rwx corresponde a todos los demás usuarios y define la accesibilidad de estos a los archivos.

Para fijar ideas un archivo que tenga los permisos rw-rw-r puede ser leído por todos los usuarios del sistema y modificado por el dueño y el grupo al cual pertenece el dueño.

Las dos columnas que siguen a los permisos indican el nombre del dueño y del grupo. Las tres columnas siguientes indican el tamaño, la fecha y la hora del archivo.

A continuación les brindamos una lista de los comandos más usuales:

| |
|---|
| Comando cat nombre del archivo: Muestra el contenido de un archivo. |
| Comando file nombre del archivo: Explica si se trata de un archivo de texto, un directorio, etc. |
| Comando mkdir nombre del nuevo directorio: Crea directorios |
| Comando rm nombre del archivo: Borra el archivo |
| Comando mv : Mover archivo |
| Comando cp : Copiar archivo |
| Comando fgrep : Busca una palabra u oración específica dentro de un archivo |
| Comando mem : Nos muestra el uso de la memoria en la máquina ordenando los procesos de mayor a menor en función del requerimiento. |
| Comando cpu : Similar a mem pero ordenando los procesos en función del uso del procesador. |
| Comando ps : Lista de procesos que están corriendo en la máquina. Si empleo el comando ps con la opción x (ps x) obtenemos un listado todos los procesos que esta corriendo nuestro usuario. ps ax muestra todos los procesos que esta corriendo en el servidor. |
| Comando kill número de un proceso: Este comando es útil para abortar un proceso que esté corriendo. El número de un determinado proceso se obtiene haciendo ps. |
| Comando man comando: Man es un manual en línea, que nos brinda toda la información disponible sobre un cierto comando, una cierta utilidad, etc. |

- Con el comando **mkdir** cree un directorio con el nombre **reacción**.
- Con la ayuda del comando **mv** cambie de nombre el directorio **reacción** por el nombre **reac_propia**.
- Copie el directorio **reac_propia** con el nombre **reac_propia2**
- Borre con un solo comando los dos directorios **reac_propia**. Utilice el comodín *.

3.3. Editores

Los editores de texto son similares al notepad. Usualmente se trabaja con Vi o con Joe.

- Vi nombre del archivo

Algunas directivas básicas para trabajar con Vi : antes de editar el archivo, presionar **Insert**. Si se desea salir del archivo o guardar los cambios presionar **Esc**. A continuación escribir: **:q!** para salir del archivo sin guardar los cambios. **:wq** para guardar el archivo .

- Joe nombre del archivo

ctr+ k+ h despliega el menú ayuda

Algunos comandos útiles son:

ctr+ k+ x guarda los cambios

ctr+ k+ q sale del archivo sin guardar los cambios

- Con el comando **joe** o **vi** cree un archivo con el nombre **reaccion.dat**.
- Cree un directorio con el nombre **reac1** y mueva el archivo **reaccion.dat** para esa nueva ubicación.
- Copie el archivo **reaccion.dat** con el nombre **reaccion2.dat** y borre el archivo **reaccion.dat**.

3.4. File Transfer Protocol (FTP)

El FTP es un protocolo de comunicación entre máquinas que permite transferir archivos a través de la red. Al igual que el SSH uno debe iniciar una sesión con la máquina a la cual uno quiere transferir archivos. Se suele proceder de acuerdo al siguiente esquema:

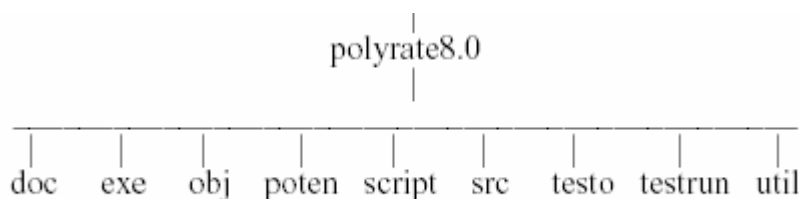
- FTP nombre de la máquina a la que se desea acceder. Ejemplo: **ftp luna.fcien.edu.uy**
- Autenticarse con nombre de usuario y contraseña.
- **Bin** (para que los archivos sean transferidos en forma binaria).

- **Put** nombre del archivo (si se desea poner un archivo que se encuentra en nuestra máquina en la máquina remota) o **Get** nombre del archivo, si lo que deseamos es traer un archivo desde la máquina remota a la máquina local.
- **Bye** para salir de la sesión de FTP.
- Vaya al menú inicio de Windows y haga clic sobre la opción ejecutar.
- Haga ftp al servidor luna.fcien.edu.uy. Autentifíquese con su nombre de usuario y contraseña.
- Transfiera el archivo **reaccion2.dat** del servidor a la máquina local.
- Transfiera un archivo local al servidor.

4. Familiarizándose con el programa Polyrate8.0

Como podrá observar a través de una sesión de SHH con el servidor luna.fcien.edu.uy, en su directorio de inicio en linux Ud. dispone de una carpeta con el nombre polyrate8.0. Dentro de ese directorio se encuentran todos los archivos necesarios para correr los ejemplos que vienen con el Polyrate y con el código fuente del programa como para adaptarlo a situaciones específicas de cada reacción (ejemplo: redimensionar la cantidad de átomos que puede manejar el programa, calcular una nueva superficie analítica, etc.)

En su directorio polyrate8.0 debería encontrar el siguiente árbol de directorios:



En la página 238 del manual del Polyrate hay un detalle de todos los archivos que debería tener una versión completa del programa y una explicación de cómo correr los ejemplos y verificar que los resultados sean los esperados.

- Navegue los directorios del programa polyrate8.0
- Vaya al directorio polyrate8.0/testrun/nh3 y corra el archivo nh3tr1.jc escribiendo el comando: **./nh3tr1.jc**

Como podrá observar para correr un archivo de entrada hace falta dos o más archivos. El archivo con extensión **.dat** es análogo al archivo **.gjf** del Gaussian. Es decir que el archivo **.dat** contiene gran parte de la información necesaria (estructuras, energías, funciones de partición, temperatura, etc) para realizar un cálculo. Dicha información podrá complementarse con archivos auxiliares como veremos en el correr de los prácticos del módulo II del Curso-Taller (archivos con extensión **.fu29**, **.fu30**, etc). A su vez se requiere de un archivo **.jc**. El archivo **.jc** es lo que llamamos un *shellscript*. Se trata de un archivo de texto ejecutable (el **.dat** es un archivo de texto no ejecutable) que contiene una serie de comandos para llevar adelante el cálculo. ¡Todo **.dat** debe tener su **.jc** asociado!

- Edite (con la ayuda del joe o del vi) el archivo de entrada **nh3tr1.dat** y el archivo ejecutable asociado **nh3tr1.jc**
- Edite el archivo de salida **nh3tr1.fu6**.
- Lea la introducción al programa disponible en la página 30 del manual.