

Cómo Trabajar Remotamente en aragorn

Aragorn es el nombre del servidor para los alumnos de electrónica/telemática. Para trabajar en aragorn, deben utilizarse programas de conexión SSH / SFTP.

Aplicaciones a utilizar

En Windows:

En caso de trabajar en Windows aplicación recomendada a utilizar para este fin es:

mobaxterm	http://mobaxterm.mobatek.net
-----------	---

Esta aplicaciones es bastante intuitiva de utilizar.

En Linux:

Si utiliza Linux, no se requieren programas externos, ya que la mayoría de las distribuciones vienen con clientes ssh ya instalados y que se ejecutan desde consola con el comando ssh. Si el comando no se encuentra en su sistema, deberá instalar un paquete de SSH Client que debería estar en el repositorio (repositorio, servidores remotos desde donde su máquina baja actualizaciones y nuevos programas).

IMPORTANTE: Recuerde que ahora el nombre de usuario de aragorn corresponde al institucional, es decir su cuenta aragorn corresponde a algo del tipo:

<Nombre>.<Apellido>@aragorn.elo.utfsm.cl



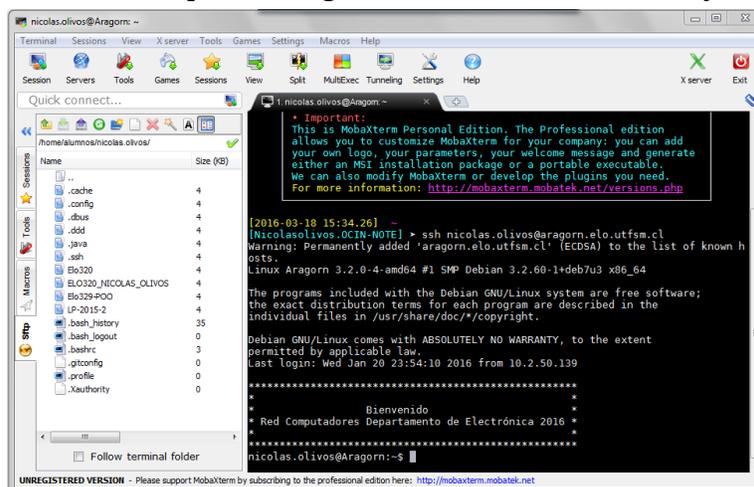
Trabajar gráficamente en aragorn

Es importante tener en cuenta que al establecer una conexión con aragorn por SSH, no podrá ver las aplicaciones gráficas que quiera correr en el servidor. Esto es porque en su forma básica SSH sólo permite transferir texto entre el computador cliente y el computador remoto.

Hace mucho tiempo que los terminales dejaron de manejar solo texto para permitir manejo gráfico (es decir ejecutar programas con ventanas, manejo de mouse, etc). La arquitectura de software para el manejo gráfico también es cliente servidor. Las aplicaciones gráficas corriendo en una máquina actúan como clientes y se conectan a un servidor gráfico en la misma máquina o en una remota y así pueden desplegar objetos gráficos en la pantalla del computador que corre el servidor gráfico. Desde la época de Unix el sistema cliente/servidor para hacer este manejo se conoce con el nombre de X Window System o sencillamente X.

Al conectarse vía ssh desde Windows o Linux, debemos informar a la aplicación nuestro interés en activar la opción de interacción gráfica (no solo texto). La aplicación que corre en el lado remoto (aragorn) en principio trataría de enviar sus salida gráfica al computador donde está (aragorn) y naturalmente arrojará un error (no tiene acceso al Display). Para permitir que esa interacción gráfica llegue hasta nuestro computador debemos solicitar a ssh traer hasta nuestro Windows o Linux la conexión del cliente gráfico (la aplicación en aragorn) y la redirija al servidor grafico (X) corriendo en nuestra máquina. Lo anterior se conoce comúnmente bajo la opción X forwarding. Así podemos ver las aplicaciones gráficas en el Display de nuestro escritorio mientras son ejecutadas computador remoto (aragorn).

En Windows: De la explicación previa se puede concluir que para ver la salida gráfica en Windows, debemos asegurarnos que éste tenga corriendo un servidor X. Afortunadamente mobaXterm ya incluye un servidor X Windows con lo cual el manejo gráfico se facilita. Lo que hace este programa es generar una consola de comandos bastante similar a la de las distribuciones Linux. Además puedes cargar directamente tus archivos y/o carpetas.



Preparado por: Tomás Girardi, Edward Alvarado; revisión 2016 Nicolás Olivos.



En Linux: Al correr ssh sólo debe agregar la opción -X.

```
$ssh -X <nombre_usuario>@aragorn.elo.utfsm.cl
```

donde <nombre_usuario> es del tipo <Nombre>.<Apellido> (como su cuenta de correo institucional).

NOTA: Para probar el uso de forwarding se recomienda correr dentro del servidor el comando \$xclock, al ejecutar este comando debiera ver una ventana con un reloj.

Subir y descargar archivos desde aragorn

Muchos alumnos suelen trabajar desde sus casas subiendo a aragorn los archivos que van desarrollando; por ejemplo, para luego compilarlos y correrlos en aragorn.

Desde Windows:

En la consola de mobaxterm usted puede ejecutar el comando scp (copia segura) y usar las mismas opciones que se explican para Linux.

Desde Linux:

Usted tiene al menos dos opciones: scp (por secure copy) y sftp (por secure ftp File Transfer Protocol). En el primer caso, corra el comando:

```
$ scp <origen> <destino>
```

donde <origen> y < destino> corresponde a el o los archivos a transferir. La ubicación local se especifica con la ruta y luego el archivo o directorio. La ubicación remota se especifica usando <nombre_usuario>@aragorn.elo.utfsm.cl:<directorío o archivo>. Ejemplos:

```
$ scp agustin.gonzalez@aragorn.elo.utfsm.cl:ramos/elo322/T1/salidaGetManual.txt .
```

Este comando transfiere el archivo salidaGetManual.txt ubicado en el directorio ramos/elo322/T1/ de la cuenta agustin.gonzalez en aragorn al directorio desde donde se ejecuta el comando. En sistemas tipo Unix, como Linux, un punto representa el directorio local.

Otro ejemplo:

```
$ scp comandoGetManual.txt agustin.gonzalez@aragorn.elo.utfsm.cl:ramos/elo322/T1/
```

Este comando transfiere el archivo comandoGetManual.txt ubicado en el directorio donde ejecuto el scp localmente al directorio ramos/elo322/T1/ de la cuenta agustin.gonzalez en aragorn. Más opciones de scp las puede estudiar ejecutando:

```
$ man scp
```

Otra opción para transferir archivos es usar sftp. Para esto debe utilizar el comando.



Profesor: Agustín González

```
$ sftp <nombre_usuario>@aragorn.elo.utfsm.cl
```

Después a través de comandos put o get se pueden transferir archivos. Más información sobre comandos de SFTP se obtiene ingresando help luego del comando previo o revisando:

<https://www.digialocean.com/community/articles/how-to-use-sftp-to-securely-transfer-files-with-a-remote-server>

Comando Script

El comando Script nos sirve para guardar en un archivo de texto toda la interacción de texto hecha en un terminal. Es útil para tener un registro de los comandos que escribimos y las respuestas recibidas en pantalla.

Su forma de uso es sencilla para inicializarlo solo ponemos script y luego la ruta del archivo donde deseamos guardar así:

```
$script archivo.txt
```

Luego para terminar de guardar en el archivo solo ejecutar el comando exit:

```
$exit
```

Todo lo que hayamos escrito, incluyendo salidas de consola entre script y exit estará guardado en el archivo que indicamos antes.

Link de interés

1- Cómo crear un túnel para acceder a páginas que solo son accesibles en la universidad con (Ej: IEEE explore para buscar papers)

Linux:

http://profesores.elo.utfsm.cl/~agv/elo322/1s10/project/reports/Enalece.web.remoto.a.taves.de.SSh_badilla_espinoza_reyes.pdf

Windows:

<http://alvinalexander.com/unix/edu/putty-ssh-tunnel-firefox-socks-proxy/>

2- Diferencias entre ssh y telnet

<http://rassler-redes.blogspot.com/2010/04/diferencias-entre-ssh-y-telnet.html>

Preparado por: Tomás Girardi, Edward Alvarado; revisión 2016 Nicolás Olivos.

