Estructura de Datos y Algoritmos
Departamento de electrónica
Universidad Técnico Federico Santa María

DEPARTAMENTO DE ELECTRONICA

Conexión a Aragorn y comandos Linux

Objetivo Principal: Conocer cómo trabajar remotamente en el servidor Aragorn y los comandos Linux potencialmente útiles para el curso.
Autor: Kevin Pizarro Aguirre

---

1. Aragorn

Las tareas del curso serán evaluadas según el funcionamiento dentro de Aragorn, un servidor basado en alguna distribución de Linux. Esto se debe hacer así puesto que C depende, en cierta manera, de los parámetros o capacidades de la máquina para obtener sus resultados, de esta forma nos aseguramos de trabajar en la misma máquina y obtener el mismo resultado.

Conexión a Aragorn

Para poder conectarnos a un servidor remoto la forma por excelencia es utilizando el protocolo SSH (Secure Shell), la cual funciona independiente del sistema operativo mediante el siguiente comando.

```
$ ssh username@aragorn.elo.utfsm.cl
```

Donde el **username** usualmente corresponde a **nombre.apellido**. Si es la primera vez que se intenta acceder al servidor desde dicha máquina entonces preguntará si se está seguro que quiere conectarse, se debe aceptar. Finalmente preguntará por la contraseña, la cual corresponde a la utilizada para el ingreso al Aula.

Una alternativa es utilizar programas que implementen SSH y SFTP en una interfaz gráfica de usuario (GUI), como por ejemplo **MobaXterm**. De esta manera se podrá abstraer de algunos de los comandos Linux, utilizar un editor de texto alternativo y otras configuraciones de manera sencilla. Se puede descargar desde el siguiente enlace. Su conexión es similar a cuando se utiliza por línea de comando pero con ayuda de la interfaz; recordamos que el host remoto corresponde a **aragorn.elo.utfsm.cl**, el nombre de usuario específico a **nombre.apellido** (generalmente) y el puerto por defecto debe ser el 22 (por protocolo SSH).

Comandos Linux

De los comandos Linux más útiles para el curso son:

Manejo de archivos locales

- ls: Para listar los archivos y carpetas en el directorio que nos encontremos.
- cd <ruta>: Para movernos a otro directorio en la ruta determinada, si no le damos la ruta regresamos a la carpeta raíz u origen.
- mkdir <dir_name>: Creamos un directorio (carpeta) con el nombre especificado.
- rm <file>: Eliminamos el archivo especificado.
- rm -r <dir_name>: Eliminamos el directorio especificado.
- cp <file1> <file2>: Copiamos el archivo 1 en el archivo 2. Ejemplo, cp archivo.c /ELO320/copia.c
- mv <file1> <file2>: Mueve el archivo 1 en la dirección del archivo 2.
- touch <file.c>: Crea el archivo vacío file.c
- more <file>: Despliega por pantalla el contenido de file.

Conexión remota, descarga y subida de archivos remotos

- ssh user@host: Nos conectamos a un servidor, explicado más detallado en la sección 1.
- scp user@host:<file1> <ruta/file2>: Copiamos un archivo 1 desde el servidor hacia nuestra maquina local con nombre file 2.

Otros

- nano <file>: Abrimos el editor de texto nano con el archivo indicado.
- vim <file>: Abrimos el editor de texto vim con el archivo indicado.
- gcc <file>: Compila el archivo de código fuente. Se le pueden añadir opciones de compilación adicionales según lo necesitado.
- ./<file>: Ejecuta el archivo ejecutable bajo el nombre indicado.
- exit: Para cerrar la sesión.

NOTA: Recordar que si se utiliza MobaXterm u otra GUI que permita conexión ssh, entonces sólo los comandos **ssh**, **gcc** y **./** son los indispensables.

Compilación y ejecución

Una vez tengamos nuestro programa escrito, nos gustaría poder ejecutarlo para ver su funcionamiento. Antes que todo se debe compilar el archivo de código fuente (archivo.c), para así generar un archivo de ejecución, el cuál por defecto se llamará **a.out**. Finalmente se deberá ejecutar el archivo de ejecución. A continuación los comandos para poder compilar y ejecutar los archivos respectivamente.

```
$ gcc archivo.c
```

Listing 1: Compilación del archivo de código fuente.

```
$ ./a.out
```

Listing 2: Ejecución del archivo ejecutable.

En el caso que se quisiera dar un nombre en específico al archivo ejecutable se agrega como parámetro de compilación -o archivo.out.

```
$ gcc archivo.c -o archivo.out
```

Listing 3: Compilación del archivo de código fuente especificando el nombre de la salida.

Luego para ejecutar se reemplaza a.out por archivo.out en el comando de ejecución. Notar que si existen problemas de sintaxis o errores similares dentro del código fuente, no se compilará correctamente y por ende no se creará un archivo ejecutable; se deberá solucionar los problemas que hubiesen hasta poder compilar sin problemas.

Configuraciones adicionales en MobaXterm

Editor de texto: Una vez iniciado MobaXterm, se deberá ir a la opción "Settings", en la opción "General" luego en la opción **Default text editor program:** se deberá cambiar al archivo ejecutable del editor de texto que se prefiera. El editor de texto por defecto será MobaText. Algunos editores de texto recomendados son *Sublime Text*, *Atom* y *Notepad++*.

Color de la terminal y fuente: Una vez iniciado MobaXterm, se deberá ir a la opción "Settings", en la opción "Terminal", luego en la opción "Color scheme" se selecciona la que nos sea más cómoda. En esa misma pestaña, en la sección **Default terminal font settings** se puede modificar la fuente de texto y su tamaño por defecto. Para modificar el tamaño momentaneamente también se puede presionar la tecla Ctrl y girar la rueda del mouse.