

## Segundo Proyecto Programado: Intérprete de Comandos

Prof. Jaime Gutiérrez Alfaro  
Arquitectura de Computadoras  
Instituto Tecnológico de Costa Rica

En el diseño de un sistema computacional, uno de los elementos que hoy en día es ampliamente investigado es la interacción humano computador. Si bien es cierto que esta línea de trabajo es desarrollada en conjunto por varias disciplinas, entre las que además de computación sobresalen las ciencias del comportamiento, diseño y ciencias cognitivas; el área tecnológica y en particular la ingeniería en computación lleva un peso especialmente importante pues suya es la responsabilidad de hacer posible vía software la interacción entre los sistemas y los cada vez más distintos dispositivos creados para interactuar con los humanos.

```
(root@localhost ~)# ping -q fa.wikipedia.org
PING text.pmtpa.wikimedia.org (208.80.152.2) 56(84) bytes of data.
^C
--- text.pmtpa.wikimedia.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 540.528/540.528/540.528/0.000 ms
(root@localhost ~)# pwd
/root
(root@localhost ~)# cd /var
(root@localhost var)# ls -la
total 72
drwxr-xr-x. 18 root root 4096 Jul 30 22:43 .
drwxr-xr-x. 23 root root 4096 Sep 14 20:42 ..
drwxr-xr-x.  2 root root 4096 May 14 00:15 account
drwxr-xr-x. 11 root root 4096 Jul 31 22:26 cache
drwxr-xr-x.  3 root root 4096 May 18 16:03 db
drwxr-xr-x.  3 root root 4096 May 18 16:03 empty
drwxr-xr-x.  2 root root 4096 May 18 16:03 games
drwxrwx--T.  2 root gdm  4096 Jun  2 18:39 gdm
drwxr-xr-x. 38 root root 4096 May 18 16:03 lib
drwxr-xr-x.  2 root root 4096 May 18 16:03 local
lrwxrwxrwx.  1 root root   11 May 14 00:12 lock -> ../run/lock
drwxr-xr-x. 14 root root 4096 Sep 14 20:42 log
lrwxrwxrwx.  1 root root   10 Jul 30 22:43 mail -> spool/mail
drwxr-xr-x.  2 root root 4096 May 18 16:03 nis
drwxr-xr-x.  2 root root 4096 May 18 16:03 opt
drwxr-xr-x.  2 root root 4096 May 18 16:03 preserve
drwxr-xr-x.  2 root root 4096 Jul  1 22:11 report
lrwxrwxrwx.  1 root root    6 May 14 00:12 run -> ../run
drwxr-xr-x. 14 root root 4096 May 18 16:03 spool
drwxrwxrwt.  4 root root 4096 Sep 12 23:50 tmp
drwxr-xr-x.  2 root root 4096 May 18 16:03 yp
(root@localhost var)# yum search wiki
Loaded plugins: langpacks, presto, refresh-packagekit, remove-with-leaves
rpmfusion-free-updates                               | 2.7 kB      00:00
rpmfusion-free-updates/primary_db                     | 206 kB      00:04
rpmfusion-nonfree-updates                             | 2.7 kB      00:00
updates/metalink                                       | 5.9 kB      00:00
updates                                                | 4.7 kB      00:00
updates/primary_db                                     73% [*****] 62 kB/s | 2.6 MB      00:15 ETA
```

Línea de comando de Linux, Bash en una terminal de GNOME.

Licencia GNU GPL v3 ZxxZxxZ @ wikipedia

La interfaz de línea de comandos (*Command Line Interface, CLI*) es sin duda alguna la forma de interacción humano computador clásica, terrorífica para un usuario general pero fantástica para una persona con conocimientos técnicos. Una CLI hace uso de dos de los elementos de entrada y salida estándar en un computador, el teclado y el monitor<sup>1</sup>.

En este proyecto programado su asignación es la implementación de una interfaz de línea de comandos, un programa donde se pueden escribir comandos (en un *prompt*<sup>2</sup>) de modo que el usuario teclea la orden que quiere llevar a cabo y al presionar la tecla «enter» el programa validará el comando y los argumentos ingresados para llevar a cabo la acción que corresponda. El objetivo de ésta trabajo es desarrollar aún más las destrezas en programación de bajo nivel, ensamblador en particular, así como otros elementos que deberá investigar por su cuenta.

## ***Descripción del software a implementar***

La aplicación a desarrollar debe ser capaz de interpretar y ejecutar los siguientes comandos: *borrar, mostrar, renombrar, copiar, comparar y salir*.

Cada uno de ellos recibirá distintos argumentos que determinarán su funcionamiento, sin embargo todos con excepción del comando salir, tendrán como argumento (opcional) común «--ayuda» que mostrará en pantalla una pequeña explicación del funcionamiento del comando, dado que este parámetro está disponible para todas las opciones de uso del intérprete, se explicará esa funcionalidad más adelante en este documento. Otra característica común en estos comandos es que el correcto funcionamiento de cada uno no deberá indicarse (obligatoriamente) mediante ningún mensaje al finalizar su ejecución sin embargo se le invita a mostrar algún mensaje que anuncie el buen funcionamiento para facilitar el proceso de revisión.

---

1 En tiempos todavía más remotos, el monitor era sustituido por una impresora como el mecanismo de salida a la hora de interactuar con un dispositivo computacional.

2 Según wikipedia: «Se llama *prompt* al carácter o conjunto de caracteres que se muestran en una línea de comandos para indicar que está a la espera de órdenes. Éste puede variar dependiendo del intérprete de comandos y suele ser configurable.  
En el *Bourne Shell* y sus derivados como *bash* (utilizados ampliamente en sistemas tipo UNIX) el *prompt*, generalmente, contiene el carácter \$ para los usuarios sin privilegios y # para el administrador, además de proporcionar más información. Por ejemplo: `jou@home-comp:/media/disk$` donde el usuario en uso es jou, el nombre del equipo es home-comp, y el directorio de trabajo es /media/disk.»

A continuación se presenta el comportamiento y sintaxis de cada uno de los comandos<sup>3</sup>.

## **Borrar**

Elimina un archivo.

### *Sintaxis:*

```
cli> borrar [--ayuda] nombre-archivo [--forzado]
```

### *Descripción:*

Este programa recibe como parámetro el nombre de un archivo (nombre-archivo) y lo elimina del sistema, pero antes debe lanzar una pregunta para verificar que el usuario está seguro de desea eliminar el archivo por completo, si la respuesta es afirmativa procede con la eliminación y en caso contrario finaliza su ejecución (vuelve al «*prompt*»). El argumento opcional --forzado tiene como objetivo omitir la pregunta de verificación, en otras palabras elimina el archivo sin solicitar la confirmación.

## **Mostrar**

Muestra en la salida estándar el contenido de un archivo.

### *Sintaxis:*

```
cli> mostrar [--ayuda] nombre-archivo
```

### *Descripción:*

Este programa recibe como parámetro el nombre de un archivo (nombre-archivo), cuyo contenido será mostrado en la pantalla (salida estándar).

---

<sup>3</sup> Para los ejemplos se utilizará como *prompt*: « **cli**> »

## Renombrar

Modifica el nombre de un archivo.

### *Sintaxis:*

```
cli> renombrar [--ayuda] nombre-archivo-original nombre-archivo-nuevo [--forzado]
```

### *Descripción:*

Los argumentos del programa son los nombres de los archivos involucrados, el archivo que debe existir en la ruta disponible al *prompt* será nombre-archivo-original y se le asignará por nombre el que se haya ingresado como argumento nombre-archivo-nuevo. Al igual que ocurre en el caso del comando borrar debe lanzarse un mensaje de verificación cada vez que se utilice este comando, para omitirlo se puede utilizar el argumento --forzado.

## Copiar

Copia el contenido de un archivo en otro nuevo.

### *Sintaxis:*

```
cli> copiar [--ayuda] nombre-archivo-original nombre-archivo-nuevo
```

### *Descripción:*

Los argumentos del programa son, el nombre del archivo que será copiado nombre-archivo-original y el nombre que tendrá el nuevo archivo por crear nombre-archivo-nuevo.

## Comparar

Toma dos archivos y los compara línea por línea.

*Sintaxis:*

```
cli> comparar [--ayuda] nombre-archivo-1 nombre-archivo-2
```

*Descripción:*

Dados dos nombres de archivo existentes, los compara línea por línea mostrando como salida los números de línea donde los archivos son diferentes, en caso de ser idénticos (por ejemplo cuando se usa el mismo archivo) tanto para nombre-archivo-1 como para nombre-archivo-2, entonces no debería imprimirse nada (pues no hay ninguna línea en la cual difieran).

## Salir

Finaliza la ejecución del programa.

*Sintaxis:*

```
cli> salir
```

*Descripción:*

Este comando no recibe ningún argumento adicional (y si recibiera alguno lo ignora e igual se ejecuta el comando).

## Funcionalidad de ayuda

El argumento opcional `--ayuda` debe hacer que se muestre un mensaje pequeño que explique el funcionamiento del comando donde se usó el parámetro. Para cada comando se contará con un archivo de texto con el nombre «comando.ayuda» (donde comando se sustituirá según corresponda por ejemplo *comparar.ayuda*), ahí en ese archivo se tendrá el mensaje de ayuda que debe desplegarse en caso de

utilizar ese argumento opcional.

## Funcionalidad de implementación opcional

Los archivos de bitácora (*log*) son utilizados en computación para ofrecer a los programadores un mecanismo para registrar los eventos más relevantes durante la ejecución de una aplicación. Una buena bitácora es importantísima principalmente para la detección de fallos, pues permite reconstruir la secuencia de pasos que pudo generar el problema en cuestión. De forma opcional se le solicita la implementación de una bitácora, en la cual registre el funcionamiento del intérprete de línea de comandos.

## Documentación

Parte fundamental de todo trabajo ingenieril es la documentación. Una buena documentación externa es concreta y precisa, útil para la persona que la escribe más que para quien la lee. Para este proyecto programado deben documentar en el código fuente y además se les solicita presentar un documento escrito o documentación externa con las siguientes características:

- Portada
- Introducción
- Análisis del problema
  - Debe responder a la pregunta: ¿Cómo lo resolvieron? e incluir como respuesta un análisis crítico de las decisiones tomadas, diseño e implementaciones realizadas . No ahonden en todos los problemas encontrados y las distintas soluciones que tuvieron que probar (para eso está la bitácora).
- Conclusiones y Recomendaciones
  - De manera muy puntual: *¿Qué pueden concluir de este trabajo luego de haberlo realizado?, Si otra persona tuviera que comenzar el trabajo desde cero, ¿Que le recomendaría?*
- Referencias

- *Asegúrese de utilizar algún estándar para escribir correctamente las fuentes de las cuales realizó consultas. Se le recomienda utilizar el estándar de APA o IEEE.*
- Anexos
  - **Bitácora de trabajo**
    - *Cada entrada de la bitácora comienza con la fecha y hora en la que se está trabajando, luego debe aparecer ¿en qué se está trabajando?, ¿qué problemas se encuentran? Y las posibles soluciones como se abarquen. En esta sección pueden colocar enlaces (links) o referencias sin necesidad de utilizar algún estándar para citarlos. La bitácora de trabajo es el cuadernillo de apuntes mientras se trabaja.*
    - *La bitácora la pueden trabajar en papel y luego adjuntarla en digital (escaneada o fotografiada).*

## ***Evaluación***

Este proyecto programado tiene un valor de 20% de la nota final del curso, el cual será distribuido en dos partes: 60% la revisión del software y 40% la documentación. Tome en cuenta que en la documentación debe incluir todo lo que le pueda ayudar durante la revisión. Lo que no esté documentado no será tomado en cuenta.

## ***Aspectos administrativos***

- Esta especificación se les está entregando el día 5 de noviembre y se contará con **2 semanas** para completarlo, por lo tanto debe estar completado para el día martes 19 de noviembre<sup>4</sup>.
- Esta asignación puede ser realizada **en parejas**. Entre los distintos grupos de trabajo pueden debatir sobre posibles formas de resolver los problemas pero las implementaciones deben ser realizadas por cada grupo de trabajo y cada uno de los integrantes debe estar preparado para defender su trabajo durante una revisión.
- La entrega del proyecto será digital, por medio de un repositorio público de **git**, de modo que no deben enviarme nada más que el URL del repositorio al correo ***jaimегutierrez.proyectos@gmail.com***. Se le sugiere utilizar **github** para manejar su software.

## ***Referencias***

[1] Prompt. (2013, 28 de octubre). Wikipedia, La enciclopedia libre. Fecha de consulta: 16:18, noviembre 4, 2013 desde <http://es.wikipedia.org/w/index.php?title=Prompt&oldid=70454663>.

---

4 Por favor indicar cualquier problema con las fechas en el mismo momento en que se le está entregando este documento.