

Sistemas Operativos

Tarea #1

Shell Linux

Escuela de Ingeniería Civil Informática
Universidad de Valparaíso

Nombre: Italo López

Correo institucional: italo.lopez@alumnos.uv.cl

31 Objetivo

En esta tarea se verá la programación a nivel de comandos en un ambiente Unix. El objetivo es familiarizarse con los comandos más usados y conocer algunos aspectos de la programación shell.

32 Instrucciones

Su trabajo deberá ser entregado en el servidor en su cuenta en GitHub.com. El nombre del repositorio debe ser SSOO-tarea01

```
+ /SSOO-tarea01
  - README.md
  - Makefile
  - src/
```

El archivo **README.md** es un archivo en markdown, donde se debe explicar el diseño de su solución, funciones utilizadas, etc. Además, debe especificar el nombre del autor y su correo institucional. En el directorio **src/** están los códigos fuentes de su proyecto. Además, debe de entregar un documento llamado “ApellidoPaterno_Nombre.pdf” evidenciando el trabajo realizado, de **NO** cumplir con el método de entrega solicitado será calificado con nota mínima.

33 Trabajo a realizar

a) Estudie y explique para qué sirven los comandos `ls`, `cat`, `chmod`, `echo`, `grep`, `cp`, `mv`, `rm` y `wc`. Dé ejemplos de uso. Para el comando `ls` averigüe para qué sirven las opciones `-l`, `-t` y `-a`. De ejemplo de uso para cada uno de esas opciones y la combinación de ellas. Ayuda: use el comando `man`, ej. `man ls`.

A continuación se presenta una tabla donde se definen los comandos solicitados y sus respectivos ejemplos.

Comando	Definición	Ejemplo
<code>ls</code>	Este comando te muestra la lista de los archivos y directorios de la carpeta que te encuentras actualmente. También es posible ver el contenido de otros directorios, escribiendo <code>ls</code> y la ruta del directorio.	<code>ls ~/Sistemas operativos/Tarea 1</code>
<code>ls -l</code>	Este comando te muestra toda la información de archivos y los directorios. Esta información incluye usuario, nombre de grupo, permisos, tamaño, fecha y hora de creación.	<code>ls -l nombredelarchivo.txt</code>
<code>ls -t</code>	Este comando ordena los archivos y carpetas de un directorio según la fecha de la última modificación.	<code>ls -t nombredelarchivo.txt</code>
<code>ls -a</code>	Este comando permite ver los ficheros del directorio, incluyendo los ficheros ocultos, los cuales son marcados con un punto(.) al comienzo.	<code>ls -a nombredelarchivo.txt</code>
Para el comando <code>ls</code> y sus variables, estas se pueden combinar de las siguientes maneras: <ul style="list-style-type: none">- <code>ls -lt</code>- <code>ls -at</code>- <code>ls -al</code>- <code>ls -lta</code>		
<code>cat</code>	Este comando te permite realizar varias acciones. La más utilizada es para visualizar el contenido de un archivo de texto directamente desde la consola. Además, con este comando puedes crear archivos, redirigir contenido, concatenar archivos.	<code>cat > nombredelarchivo.txt</code>
<code>chmod</code>	Este comando se utiliza para cambiar los permisos de archivos y directorios.	<code>chmod u = rwx nombreadelarchivo.txt</code>
<code>echo</code>	Este comando sirve para imprimir una cadena de strings en la consola de comandos.	<code>echo *.jpg</code>
<code>grep</code>	Este comando te permite buscar palabras en un archivo de texto. Existen varias opciones que te permiten realizar la búsqueda sin tener en cuenta las mayúsculas y minúsculas, realizar un conteo de las veces que una palabra aparece en el archivo o también para encontrar una palabra en un conjunto de archivos.	<code>grep -c palabra nombreadelarchivo.txt</code>

cp	Este comando te permite realizar una copia de un archivo desde el directorio actual hacia un directorio diferente.	cp archivo.txt /home/usuario/Escritorio/Sistemas Operativos/
mv	Este comando puede ser utilizado para mover o para cambiar el nombre de algún archivo.	mv archivo.txt /home/usuario/Escritorio/Sistemas Operativos/
rm	Este comando se utiliza para eliminar archivos y directorios.	rm archivo.txt
wc	Este comando se utiliza para contar palabras, letras, bytes, líneas de un archivo.	wc -m archivo.txt

b) Explique qué son los metacaracteres y dé ejemplos de uso de ellos.

Los metacaracteres son un conjunto de caracteres que se encuentran reservados en el sistema para realizar ciertas funciones en especial en el shell. A continuación se dispone de una tabla explicando algunos de estos metacaracteres y dando un respectivo ejemplo.

Metacaracteres	Significado	Ejemplo
*	Es un comodín de sustitución, sustituye a 0 o más caracteres.	*.jpg
?	Es un comodín de sustitución, sustituye a 1 carácter.	ls ejem?lo.txt
[]	Es un comodín de sustitución, sustituye todo carácter que se encuentre dentro de los corchetes.	ls [12]archivo.txt
<	Hace una redirección de la entrada. Lee la entrada estándar de un fichero.	

c) Explique en qué consiste la expansión por paréntesis de conjunto (Brace Expansion). Con esta herramienta, resuelve el siguiente problema:

En un directorio, se quieren crear subdirectorios para que almacenen respaldos diarios de todo un año. Debe tener la siguiente estructura :

```
directorio_actual/  
+ 2021-01-01/  
+ 2021-01-02/  
+ 2021-01-03/  
.  
.  
.  
+ 2021-09-18/  
.  
.  
.  
+ 2021-10-18/  
.  
.  
.  
+ 2021-12-31/
```

Suponga que todos los meses tienen 31 días. Debe ejecutar UN sólo comando para crear la estructura de directorios solicitada.

La expansión por paréntesis de conjunto (Brace Expansion) es una técnica utilizada para generar listas de cadenas en la línea de comandos de Linux. Este mecanismo tiene distintos métodos de uso. Primero, si ingresas una lista separada por comas, esta será impresa en la línea de comandos tomando cada elemento de la lista por separado. Por ejemplo, si ingresas **echo {1,2,3,4}**, se imprimirá el contenido de la lista sin las comas, es decir, 1 2 3 4. Además de esto, también si se desea imprimir una serie de números o letras, dentro de los paréntesis se debe ingresar el valor inicial y el valor final, separados por dos puntos. Por ejemplo, si ingresas **echo {1..5}** se imprimirán los números desde el 1 al 5.

También se pueden crear carpetas dentro de directorios en específico. Por ejemplo, si deseas crear 3 carpetas en el directorio llamado 'SistemasOperativos', lo que debes ingresar en la línea de comandos es **mkdir ~/SistemasOperativos/{uno,dos,tres}**.

Considerando lo mencionado anteriormente, para crear la estructura de directorios solicitada, se debe utilizar el siguiente comando: **mkdir 2021-{01..12}-{01..31}**.

d) La interconexión de comandos a través de *pipes* permite construir, de forma muy simple, nuevas herramientas. Como ejemplo considere los comandos **ls** y **wc**, que interconectados permiten contar archivos del directorio actual: **ls | wc -l**. Mediante el uso de pipes resuelva:

1. Mediante **grep**, encontrar archivos cuyo nombre contenga el carácter **i** en el directorio **/bin**.

El comando que se debe utilizar es: **ls ~/SistemasOperativos/ | grep i**

2. Contar los archivos con una secuencia de permisos **r-x** en los directorios **/bin** y **/usr/bin**.

El comando que se debe utilizar es: **ls -l ~/bin ~/usr/bin | grep -c r-x**

e) Las variables de ambiente definen aspectos del entorno de programación, y los comandos `set` y `echo` (mediante el metacaracter `$`) permiten ver su contenido.

1. Investigue el uso de las variables `HOME`, `SHELL`, `PATH`, y `PWD`. ¿Cómo se puede visualizar su contenido?

La variable de entorno `HOME` hace referencia a la carpeta personal de usuario, para un *usuario*, su carpeta home sería `/home/usuario`.

La variable de entorno `SHELL` contiene el nombre de la shell interactiva que se está ejecutando, como lo es `bash`.

La variable `PATH` contiene algunas de las rutas de ejecución más importantes. Estas rutas son las predeterminadas que utiliza el shell para buscar un comando cuando se escribe en el terminal.

La variable de entorno `PWD` es la que contiene la ruta al directorio de trabajo actual.

Para visualizar el contenido de todas estas variables de entorno se utiliza el comando `echo $home`, `echo $shell`, `echo $path`, `echo $pwd`.

2. Investigue cómo se puede modificar el valor de una variable de ambiente (ayuda: investigue el operador `'='` y `export`).

Mediante la utilización del comando `export` y el operador `"="` es posible cambiar el valor de una variable de entorno. Por ejemplo, para realizar un cambio de hora se debe cambiar el valor de la variable `TZ` y exportar una nueva zona horaria de la siguiente manera : `export TZ = "US/Pacific"`. Posteriormente podrías verificar este cambio utilizando el comando *date*.

3. Ejecutando el comando *bash* dentro de la línea de comandos se crea un sub-shell, ¿Cómo afecta esto las variables de ambiente? ¿Cuál es el efecto de `export`? Explique y dé ejemplos

f) El intérprete de comandos `bash` es también un lenguaje de programación, con estamentos de control de flujo como `for`, `while`, etc. El código fuente a menudo se llama *script*. Si el archivo que contiene el script se llama `ejemplo.sh`, el comando `chmod +x ejemplo.sh`, le da permisos de ejecución al archivo `ejemplo.sh`.

Implementar un script `BASH` que liste cada argumento de entrada por separado, incluyendo el nombre del script. Además, debe mostrar su `PID` y mostrar las 10 primeras líneas del archivo `/proc/PID/status`.