



LICENCIATURA EN FÍSICA  
DEPARTAMENTO DE FÍSICA

FÍSICA COMPUTACIONAL 1

---

### Reporte 4

Alumna:  
Brambilla Zamorano Fátima Fernanda

---

Fecha:  
28/02/18

# 1 Introducción

Los sistemas operativos *UNIX* y otros sistemas derivados como lo son Linux y macOS se apoyan con un interprete de comandos llamado *Shell*, el cual juega un papel de intermediario entre el usuario y el sistema operativo. Y aunque hay toda una familia de comandos de interpretación, no mencionaremos muchos en esta ocasión, ya que nos limitaremos a trabajar con el interprete: */bin/bash* que viene por default en los sistemas previamente mencionados.

# 2 Fundamentos

Los fundamentos para esta cuarta actividad son los comandos que se enlistaran más adelante, ya que serán necesarios durante la misma, y en las siguientes actividades por venir:

- **cat:** Este comando concatena archivos y los muestra en la salida estándar. Combinado con distintas opciones, ejecuta diferentes acciones, por ejemplo:
  - **-b:** omite los números de línea para los espacios en blanco en la salida
  - **-e:** un carácter \$ se mostrará al final de cada línea anterior a una nueva línea
  - **-E:** muestra un \$ (símbolo de dolar) al final de cada línea
  - **-n:** numera todas las líneas en la salida
  - **-s:** Si la salida tiene múltiples líneas vacías, las sustituye con una sola línea vacía
  - **-T:** muestra los caracteres de tabulación en la salida
  - **-v:** los caracteres no mostrados se muestran

No obstante, estas no son las únicas acciones que se pueden llevar a cabo con este comando.

- **echo:** El comando echo muestra la cadena dada en la entrada en el output estándar.
  - **-e:** activa la interpretación de los caracteres alternativos listados debajo
  - **-E:** desactiva la interpretación de esas secuencias en cadenas
- **grep:** El comando grep selecciona y muestra las líneas de los archivos que coincidan con la cadena o patrón dados. Este comando tiene varias opciones, a continuación se muestran algunas:
  - **-c:** Muestra la cuenta de líneas coincidentes
  - **-e:** busca un patrón

- -h: muestra las líneas coincidentes pero no los nombres de archivo
- -r: lee iterativamente todos los archivos en los directorios y subdirectorios encontrados
- -w: coincidencias en palabras completas únicamente
- less: El comando less se usa para mostrar texto en la pantalla de la terminal. Solo muestra el texto del archivo dado, no se puede editar o manipular.
- ls: Este comando en lista los archivos y directorios en el directorio de trabajo actual.
- wc: Esta instrucción sirve para contar líneas, palabras y caracteres que contiene un archivo.
- chmod: Este comando nos permite alterar o cambiar los permisos de acceso de un archivo o directorio. Este comando utiliza números del cero al siete para editar los permisos, y la acción de cada uno de los números se puede ver en la siguiente tabla:

#	Permiso de Archivo
0	ninguno
1	sólo ejecutar
2	sólo escritura
3	escritura y ejecución
4	sólo lectura
5	lectura y ejecución
6	lectura y escritura
7	todos los permisos

### 3 Procedimiento y Análisis de Datos

En primer lugar descargamos un archivo *script1.sh* de la pagina de pbworks del grupo, el cual nos ayudaría a descargar automáticamente los datos meteorológicos de un año de alguna ciudad, desde la pagina de la universidad de Wyoming. Para esto, editamos el archivo *script1.sh* un poco, cambiando la estación de la cual buscaría la información, así como el código del programa, pues el archivo estaba hecho para descargar los datos de cinco años, pero en nuestro caso solo trabajaríamos con los dados de doce meses. Y una vez editado el archivo, cambiamos los permisos para el usuario en la terminal, pues el archivo solo podía ser leído y editado, sin embargo, no estaban los permisos para correrlo, por lo que nos apoyamos en el comando *chmod* para cambiar los permisos.

Al ejecutar el archivo, este empezó a descargar los datos, creando doce archivos nuevos en un carpeta, cada archivo correspondiente a un mes del año. Por medio del comando *less* exploramos los datos de uno de los archivos, notando

lo amplio que es cada uno de ellos, y con el comando *cat* vimos su contenido. El siguiente paso fue aprender a manejar el comando *grep* el cual funciona para hacer una extracción de datos, o buscar ciertos datos de un documento. Pero antes de hacer alguna extracción de los datos que obtuvimos de la pagina de la universidad, debimos asegurarnos de que todos los archivos descargados eran de cierto tipo, para lo cual nos apoyamos en el comando *file*. Una vez que se verifico que todos los archivos eran en efecto del mismo tiempo, utilizamos el comando *cat* para concatenar la información de los doce archivos, y ubicarla en uno solo, utilizando el comando de la siguiente manera *cat sounding\* > sondeos.txt*, quedando como nombre del archivo: *sondeos.txt* Enseguida utilizamos el comando *grep* para filtrar los datos que queríamos conservar del archivo *sondeos.txt* recién creado, por medio del siguiente comando: *egrep -v 'PRES/hPa' sondeos.txt | egrep '[estación de servicio y palabras clave]' > df2017.csv* el nuevo archivo creado resulto ser únicamente ciertas lineas de los archivos de datos de cada mes.

## 4 Resultados

El primer resultado obtenido fue después de correr el script que descargamos desde la pagina de actividades del grupo, pues nos cargo doce archivos, uno por cada mes del año, a la carpeta de trabajo en donde se encontraba dicho script. De ahí, ejecutamos el comando *ls -alg* para ver los archivos de la carpeta de trabajo, donde a su vez podíamos observar los que en efecto teníamos varios archivos nuevos en la carpeta, siendo estos el script y los doce archivos de datos meteorológicos. El siguiente resultado que menciono es después de completar hasta el decimoquinto paso de la lista de acciones que debíamos ejecutar de acuerdo a la actividad:

[illegible]

En la imagen anterior se pueden apreciar los datos finales en la carpeta de trabajo, siendo estos doce archivos del tipo *ASCII*, tres scripts del tipo *.sh*, un archivo de texto y un archivo *.csv*

## 5 Conclusiones

Al finalizar esta práctica aprendimos a hacer varias cosas nuevas, entre ellas el manejo de nuevos comandos que nos facilitan interactuar con el sistema y la

edición de archivos desde la terminal. Además de esto, se nos introdujo el editor de emacs de una forma, para la edición de archivos y creación de programas, así como algunos comandos que se utilizan dentro del propio editor, para poder trabajar con fluidez.

## 6 Bibliografía

1. hscripts.com . (s.f.). Obtenido de <https://www.hscripts.com/es/tutoriales/linux-commands/passwd.html>

## 7 Ápendice

1. ¿Qué fue lo que más te llamo la atención en esta actividad?  
Lo que más llamo mi atención durante esta práctica fue la introducción a emacs y a los comandos de /bin/bash.
2. ¿Qué consideras que aprendiste?  
Considero que aprendí un poco sobre como utilizar el editor de emacs, y es algo que me gustaría reforzar más adelante
3. ¿Cuales fueron las cosas que más se te dificultaron?  
Los comandos de /bin/bash, ya que son unos cuantos los que utilizamos y cada uno tiene varias opciones.
4. ¿Cómo se podría mejorar esta actividad?  
quizás con una explicación más detallada de los comandos nuevos, antes de empezar a trabajar con ellos.
5. En general, ¿cómo te sentiste al realizar esta actividad?  
La palabra más apropiada parece ser "perdida", en algunos momentos no sabía que estaba ocurriendo, así que tuve que recurrir bastante a mis compañeros para algo de orientación.