

Práctico 0: ShellScripting

Sistemas Operativos - 2019

Según Wikipedia, el Shell es *el programa informático que provee una interfaz de usuario para acceder a los servicios del sistema operativo*. Comenzaremos la materia familiarizandonos con esta herramienta, que usarán el resto de sus carreras, y con las llamadas de sistema más comunes que pueden acceder.

Este práctico consiste de una serie de ejercicios que se resuelven con **una** línea de comando ejecutada en la consola. En algunos casos, tendrán que clonar este repositorio en sus computadoras para utilizar los archivos dados y comprobar si el comando que proponen funciona.

Tendrán que **investigar por su cuenta** cómo funcionan los distintos comandos (como grep, cat, sort, head), pipes | y redirecciones ><. Les dejamos un pequeño machete como ayuda (*Bite_Size_Linux.pdf*). Otro recurso que les puede ser de utilidad es el sitio explainshell.com, para comandos que encuentren online y no sepan qué hacen.

Ejercicios

1. ¿Qué modelo de procesador tiene tu equipo? Cuando necesitamos un dato del equipo, como por ejemplo el modelo del procesador, podemos buscar en el archivo /proc/cpuinfo. Sin embargo, este comando nos devuelve un listado muy largo y tienen que encontrar la forma de buscar sólo la información necesaria. Ayuda: ver la definición y el uso de los comandos cat y grep.

```
grep 'vendor_id' /proc/cpuinfo ; grep 'model name' /proc/cpuinfo ; grep 'cpu MHz' /proc/cpuinfo
```

con este comando respondo que MODELO de procesador es?

```
cat /proc/cpuinfo | grep "model name" //este comando me parece mejor
```

2. Si la computadora tiene más de una unidad de ejecución (multi-core) seguramente en el punto anterior se repitió más de una vez el modelo del procesador. Averiguar cómo usar wc para poder determinar cuántas unidades de ejecución tiene el procesador, aprovechando los comandos utilizados en el ejercicio 1.

```
nproc -all ///pero con este comando no uso el wc
```

```
lscpu | grep 'CPU(s)' ///este me dice pero dice 0-3 a pesar de que son 4
```

```
cat /proc/cpuinfo | grep "cpu cores" | wc -l ///para numero de nucleos
```

Obtener el número de cpus/cores físicos:

```
~$ lscpu -p | egrep -v '^#' | wc -l
```

4

Obtener el número de cpus lógicas (incluyendo aquellos cores con HT):

```
~$ lscpu -p | egrep -v '^#' | sort -u -t, -k 2,4 | wc -l
```

8

3. Descarguen en libro Alicia en el país de las maravillas en txt (puede ser de [este link](#)), reemplacen todas las ocurrencias de Alice/Alicia por su nombre, y guarden el archivo como NOMBRE_en_el_pais_de_las_maravillas.txt o NOMBRE_in_wonderland.txt dependiendo del idioma en que lo encuentren. Sí, todo con un único comando, incluso la descarga del archivo. Puntos extra por borrar el archivo que original que descargaron.

```
wget -O [filename] [URL]
```

```
curl -o ISABEL_in_wonderland.txt https://www.gutenberg.org/files/11/11-0.txt
```

```
sed -i 's/Alice/Isabel/g' 's/ALICE/ISABEL/g' ISABEL_in_wonderland.txt
```

```
rm ISABEL_in_wonderland.txt
```

profe: Si hacés cat alicia.txt | wc va a tomar la salida de cat alicia y le va a aplicar el comando wc

este es el que hace todos los pasos:

```
wget https://www.gutenberg.org/files/11/11-0.txt && cp 11-0.txt ISABEL_in_wonderland.txt  
&& sed -i 's/Alice/Isabel/g' ISABEL_in_wonderland.txt && sed -i 's/ALICE/ISABEL/g'  
ISABEL_in_wonderland.txt && rm 11-0.txt
```

directamente descarga con el nombre:

```
wget -O ISABEL_in_wonderland.txt https://www.gutenberg.org/files/11/11-0.txt && sed -i  
's/Alice/Isabel/g' ISABEL_in_wonderland.txt && sed -i 's/ALICE/ISABEL/g'  
ISABEL_in_wonderland.txt | rm ISABEL_in_wonderland.txt
```

lo quise hacer todo junto y no pude : curl -o ISABEL_in_wonderland.txt
https://www.gutenberg.org/files/11/11-0.txt sed -i 's/Alice/Isabel/g' 's/ALICE/ISABEL/g'
ISABEL_in_wonderland.txt rm ISABEL_in_wonderland.txt

tuberias???

consulta:

grep alice <archivo>/// paara ver que no hay ocurrencias en minuscula

4. El archivo weather_cordoba.in contiene mediciones meteorológicas realizadas en un día en Córdoba. Las primeras 3 columnas corresponden al año, mes y día de las mediciones. Las restantes 6 columnas son la temperatura media, la máxima, la mínima, la presión atmosférica, la humedad y las precipitaciones medidas ese día. Ordenar los días según su temperatura máxima y calcular máximo y mínimo. Puntos extra por mostrar en la pantalla sólo la fecha de dichos días (un comando para cada día). Tengan en cuenta que para evitar los números reales, los grados están expresados en décimas de grados (por ejemplo, 15.2 grados está representado por 152 décimas). La presión también ha sido multiplicada por 10 y las precipitaciones por 100, o sea que están expresadas en centésimas de milímetro.

```
awk '{ print $4, $3 }' weather_cordoba.in sort-bdf weather_cordoba.in | uniq -ic | sort -nr
```

```
sort -nr -k 5 weather_cordoba.in |...
```

solo muestra una linea:

```
awk '$5 > max { max=$5; linea=$0 }; END { print max, linea }' weather_cordoba.in ls
```

Puntos extra por mostrar en la pantalla sólo la fecha de dichos días (un comando para cada día ??????????????)

mas entendibleeeeeee

Temperatura maxima:

```
sort -nr -k 5 weather_cordoba.in | head -n 1 | awk '{print $1,$2,$3}'
```

Temperatura minima:

```
sort -nr -k 5 weather_cordoba.in | tail -n 1 | awk '{print $1,$2,$3}'
```

Todo junto:

```
echo "Temperatura max:" && sort -nr -k 5 weather_cordoba.in | head -n 1 | awk '{print $1,$2,$3}' && echo "Temperatura min:" && sort -nr -k 5 weather_cordoba.in | tail -n 1 | awk '{print $1,$2,$3}'
```

consulta lab: concatenar dos comandos como echo "lala" ; echo "lolo" se sigue considerando como dos comandos separados, ya que el ; usando de esa manera es lo mismo que poner cada comando en una línea distinta. Sin embargo, si concatenan dos comandos con el operador && (busquen para qué sirve) sí se considera que el resultado es un único comando, ya que el && va a tener un efecto en el resultado final.

5. El archivo atpplayers.in es un listado por orden alfabético de jugadores profesionales de tenis. El nombre del jugador viene acompañado de una abreviatura de su país, el número que ocupa en el ranking, su edad, su puntaje y el número de torneos jugados en el último año. Ordenar el listado de jugadores según la posición en el ranking.

```
sort -n -k 3 atpplayers.in
```

6. El archivo superliga.in contiene datos con el siguiente formato: nombre de equipo sin espacios, puntos, partidos jugados, partidos ganados, partidos empatados, partidos perdidos, goles a favor y goles en contra. Ordenar la tabla del campeonato de la Superliga según la cantidad de puntos, y desempatar por diferencia de goles. Ayuda: a todo lo que ya saben del ejercicio anterior, agreguen el uso del comando awk.

```
sort -nr -k 2 superliga.in
```

desempatar por diferencia de goles comando final:

```
awk '{print $0, $7 - $8}' superliga.in | sort -nr -k 2 -k9 | awk '{print $1,$2,$3,$4,$5,$6,$7,$8}'
```

comparar: con awk también calcular la suma o diferencia de otras columnas, por si les sirve pensar en esos términos.

```
awk [comando] [archivo] | sort []
```

problemas menores:

1) supongamos que la columna "diferencia de goles" ya está incluida en el archivo. El problema se reduce entonces a ordenar por una columna (cantidad de puntos) y desempatar por otra (diferencia de goles). Como mencionó Matías, ya tienen el comando para hacer esto.

2) Ahora solo necesitamos agregar la columna "diferencia de goles" al archivo original, total el punto 1) ya está solucionado. Acá entra en juego awk.

Por último encadenamos las dos soluciones, una atrás de la otra, con un pipe. No considero casos particulares como "busco los equipos con misma cantidad de puntos y los ordeno". Directamente ordeno todo. Sort ya hace todo por mi.

P.D: quizás haya que borrar la columna que agregaron de diferencia de goles. No es difícil: un pipe más.

7. ¿Cómo ver la MAC address de nuestro equipo? Investiguen el comando ip. En la manual de grep van a encontrar la especificación de muchas operaciones, por ejemplo -o, -i, y muchas más. Algo muy utilizado son las expresiones regulares para realizar una búsqueda. En el manual de grep tienen un apartado donde explica su uso. Con esta información deberían poder construir una secuencia de comandos de shell para imprimir por consola la MAC address de su interfaz ethernet.

ifconfig -a //lo hace pero es un comando viejo

curl ifconfig.me //muestra un numero no estoy segura

ifconfig | grep ether ///muestra dos direcciones

La siguiente es la que me parece que debería resultar pero no :(

ifconfig eth0 | grep -o -E '([[:xdigit:]]{1,2}:){5}[[:xdigit:]]{1,2}'

ip addr | grep -C1 "link/ether"

8. Supongamos que bajaron una serie de televisión completa con subtítulos (de forma completamente legal, por supuesto). Sin embargo, los subtítulos tienen el sufijo `_es` en el nombre de cada archivo y para que puedan ser reproducidos en el televisor, que nunca fue demasiado smart, el archivo de subtítulos tiene que tener exactamente el mismo nombre que el archivo de video con la extensión `srt`. La serie tiene más de 100 capítulos, imposible realizar los cambios uno a uno.
 - a. Para emular la situación, crear una carpeta con el nombre de la serie y dentro de ella crear 10 archivos con un nombre como `fma_S01EXX_es.srt`, donde `XX` es el número de capítulo, desde 1 hasta 10.
 - b. Con un segundo comando (el que usarían realmente), cambiar el nombre de cada archivo sacando el sufijo `_es`.

Ayuda: Usar un ciclo `for`. Se pueden hacer ciclos en una única línea. Puntos extra si los archivos de video no siguen un patrón, sino que pueden llamarse de cualquier forma.

1. [OPCIONAL] El comando `ffmpeg` sirve para editar streams de video y audio desde la consola, de forma muchísimo más rápida que otros editores de video. Tienen que descubrir cómo realizar las siguientes tareas:
 1. Recortar un video. Pueden usar `SimpleScreenRecorder` (linux) para grabar un pequeño video de prueba. Luego, usen `ffmpeg` para sacarle los primeros y los últimos segundos en donde se ve el la pantalla como inician y cortan la grabación
 2. Mezclar streams de audio. Graben un pequeño audio de voz, y descarguen de internet alguna pista de sonido que quieran poner de fondo. Usen `ffmpeg` para superponer las dos pistas. Tomen como ejemplo los podcasts de Nico!

```
for var in $(cat $my_file); do
```

```
    echo " $var"
```

done

comando:

```
for f in *; do mv "$f" "${f%_es.str}.str"; done
```

consulta:

```
for i in {0..9}; do echo $i; //ese ; es la sintaxis del for asi que si valeee
```

Entrega

La resolución de este práctico es **individual**. Deben entregar un archivo .sh con una línea por comando a través de [este formulario](#). El práctico no se evalúa con nota, pero debe estar aprobado para poder promocionar o regularizar la materia.

consulta del labo:

concatenar dos comandos como `echo "lala" ; echo "lolo"` se sigue considerando como dos comandos separados, ya que el `;` usando de esa manera es lo mismo que poner cada comando en una línea distinta. Sin embargo, si concatenan dos comandos con el operador `&&` (busquen para qué sirve) sí se considera que el resultado es un único comando, ya que el `&&` va a tener un efecto en el resultado final.

el program counter apunta a la instrucción actual en ejecución, en la sección de texto en la memoria del programa donde se pone todo el código a ejecutar

el stack pointer apunta a la punta del stack, que es la memoria del proceso donde se guardan los parámetros, valores de retorno y variables automáticas de las funciones a medida que se llaman

la forma tradicional de renombrar archivos en linux es con mv: mv nombre-feo nombre-lindo