

**105.1 Scripts**

Un fichero, que nos permite a nosotros como usuarios ejecutar múltiples tareas o acciones por medio de una acción. Ejemplos de lenguajes de programación:

BASH / SH / DASH / C++

.EXE

**Primer ejemplo de script:**

cat Script\_Variable.sh

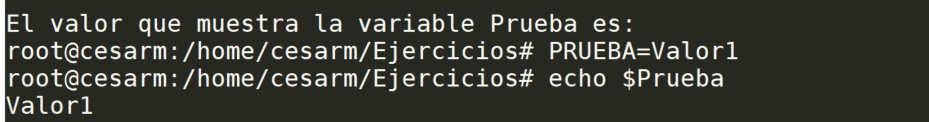
#!/bin/bash

echo "El script que imprime el valor de la variable PRUEBA"

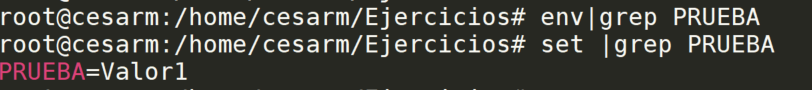
echo " "

echo "El valor que muestra la variable Prueba es:" $PRUEBA

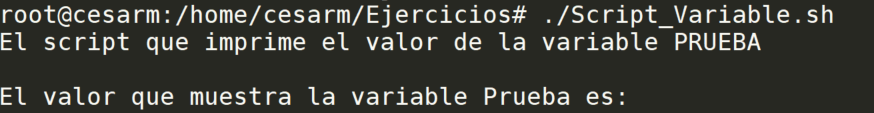
Ahora vamos a ver una variable definida localmente:



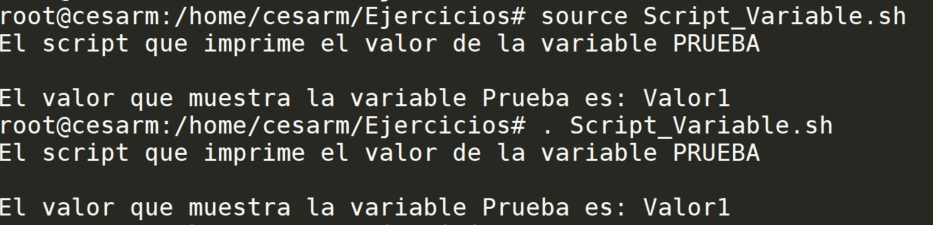
Con el primer comando ENV no se nos muestra información porque la variable no a sido exportada aún, más el segundo SET nos muestra tanto variables locales de ambientes como las exportadas de usuario.



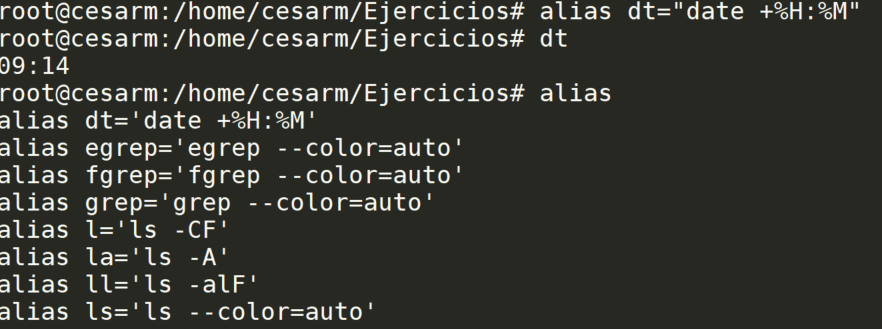
El valor no es encontrado porque no encontré la variable prueba:



Hay una forma de ejecutar el script en la sección local y es por eso por lo que existe el comando **source o punto espacio**, ejemplo:



Recordando el comando alias nos permite crear un atajo como el siguiente ejemplo:



Así como Alias crea rutinas en nuestro ambiente vamos a ver algunas otras utilidades. El comando function normalmente es utilizado para crear una rutina de comandos y podríamos utilizar if, loop, while que estudiaremos más adelante.

function funcion1 {

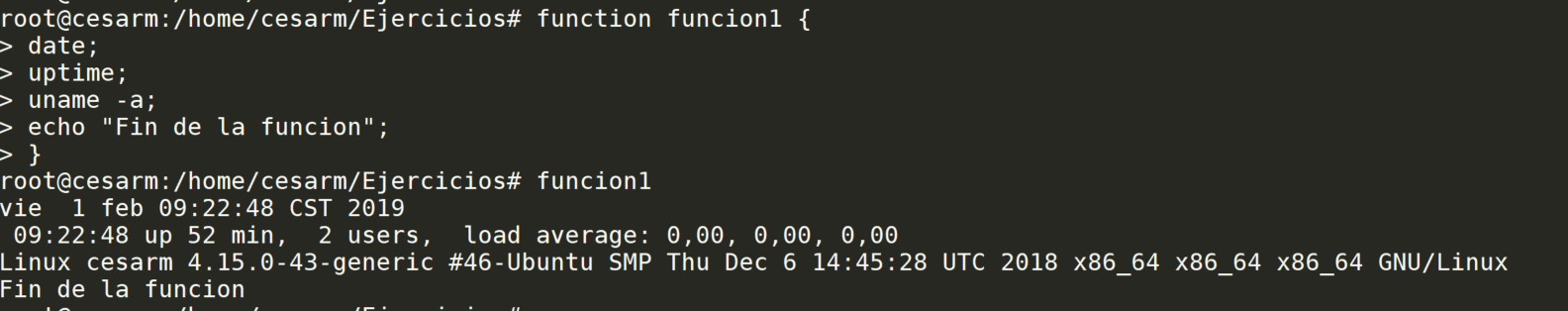
> date;

> uptime;

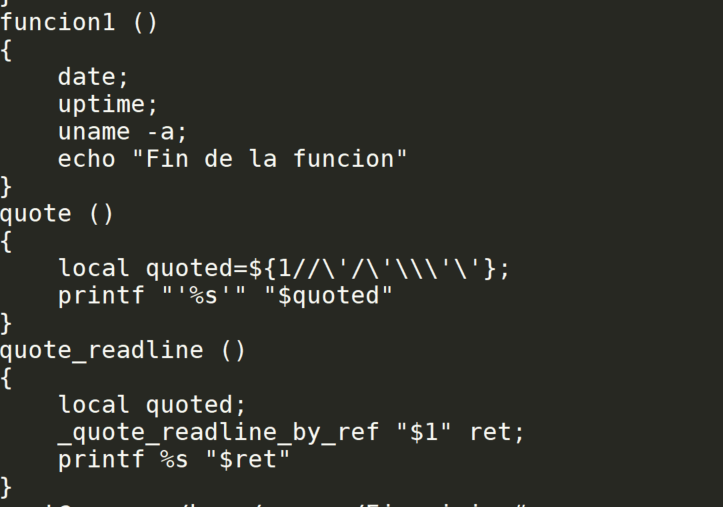
> uname -a;

> echo "Fin de la funcion";

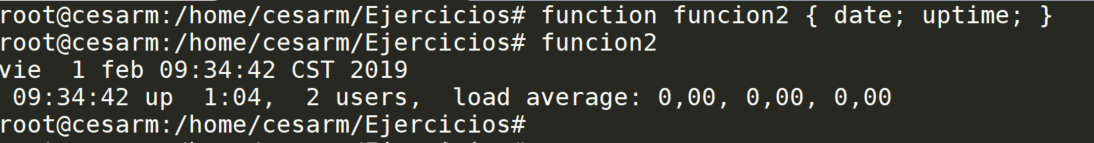
> }

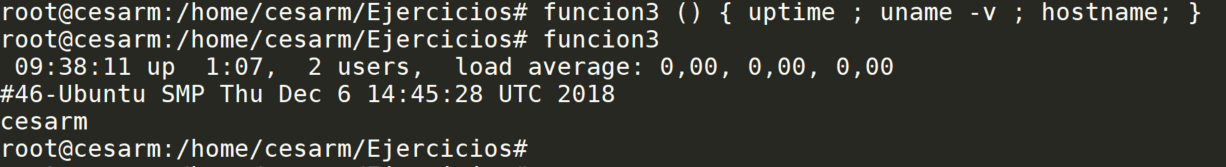


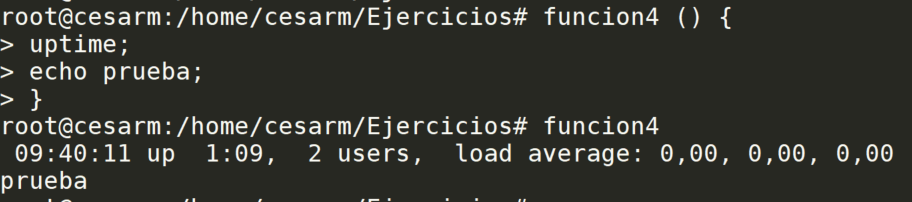
¿Como veo la definición de esta función? Digitamos el comando set y al final no debería aparecer en algún momento la funcion que definimos:



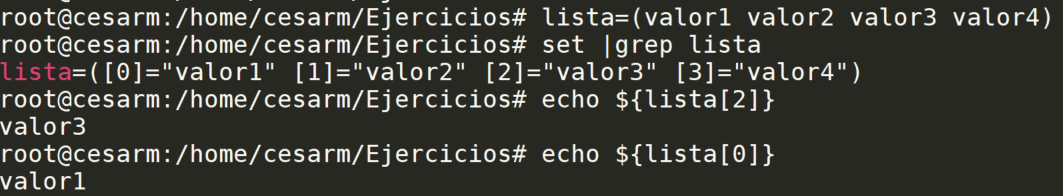
Ahora a hacer otros ejemplos de una función como se muestra a continuación:

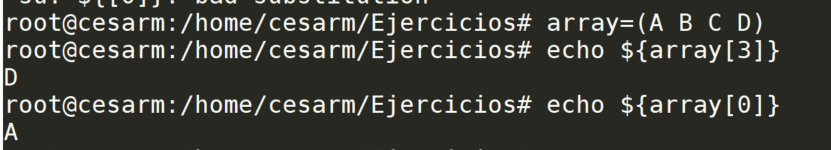




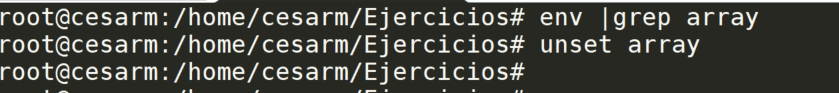


También podemos crear una lista de valores o array como en se nos muestra a continuación:



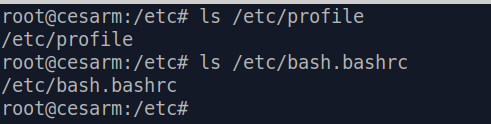


Ahora si queremos removerlo utilizamos el siguiente comando:

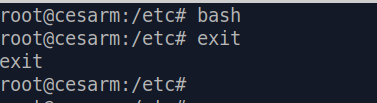


**113. 105.1 Personalización y Uso del Ambiente Shell - Archivos de configuración**

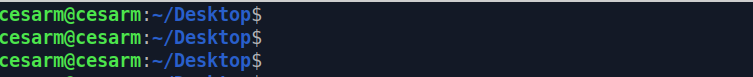
Vamos a ver los archivos que definen las configuraciones de todos los usuarios:



Si digitamos bash abrimos una nueva sesión de bash y luego con el comando exit salimos de esta sesión, esto no incluyo un procedimiento de logearse en el Shell.



Lo mismo sucede de igual forma si abrimos cualquier nueva sesión de terminal:

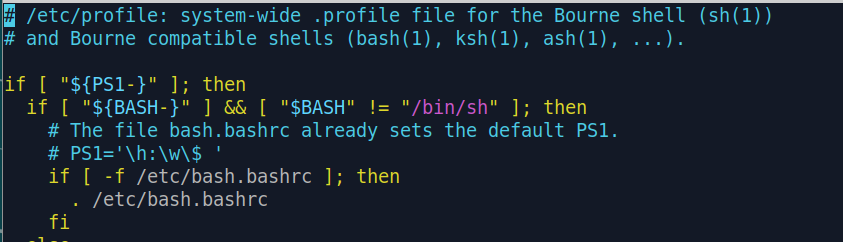


Siempre que comenzamos un nuevo login estamos abriendo un nuevo Shell o bash; mas no siempre que estoy abriendo un nuevo bash estoy logeandome.

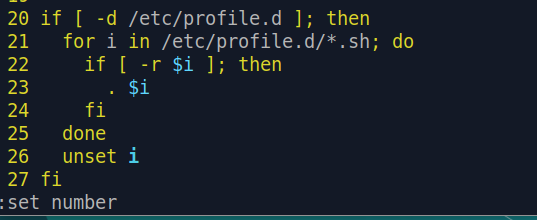
**etc/profile:** Es usado cuando cualquier usuario del sistema hace un procedimiento de logearse; tanto para interfaz gráfica como interfaz de solo texto.

**etc/bash.bashrc:** Es aplicado cuando se abre una nueva terminal, una nueva sesión de bash o Shell.

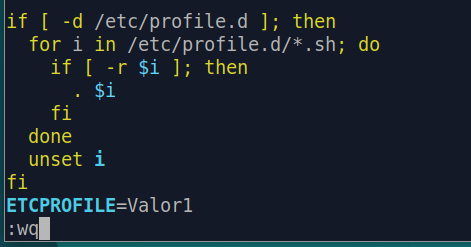
**vi /etc/profile:** Vamos a tener aquí adentro varias rutinas que ya fueron implementadas en el sistema. La primera no dice que si encuentra el fichero bashrc que ejecute el mismo bashrc:



**vi /etc/profile:** Otra rutina que tenemos es que si yo tengo el /etc/profile.d ejecute todo lo que este dentro de este perfil:

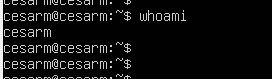


Estos son algunos ejemplos de scripts que están corriendo el sistema. Vamos a hacer un ejemplo con una variable; nos logeamos como root (raíz).

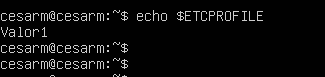


Si abrimos una nueva terminal y ejecutamos echo $ETCPROFILE, no va a encontrar nada porque no hicimos un procedimiento de login. 

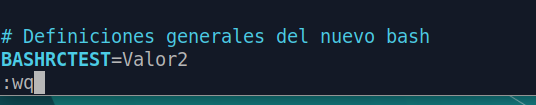
Ahora nos logeamos en la interfaz de texto como usuario local:



Ahora si se nos a reflejar el valor1 como se muestra a continuación:



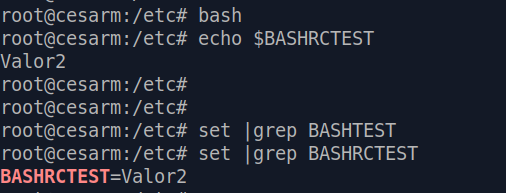
Vamos a editar el directorio /etc/bash.bashrc al final del documento como se muestra a continuación:



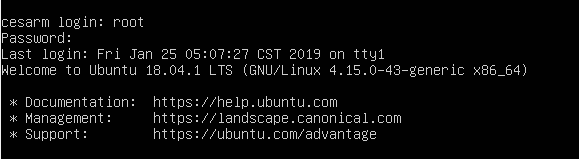
De momento no vamos a encontrar nada porque no hemos iniciado una nueva sesión bash o del Shell:

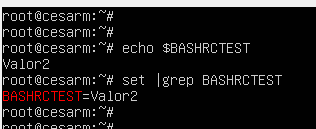


Vamos a realizar la prueba luego de abrir una nueva sesión de Shell:

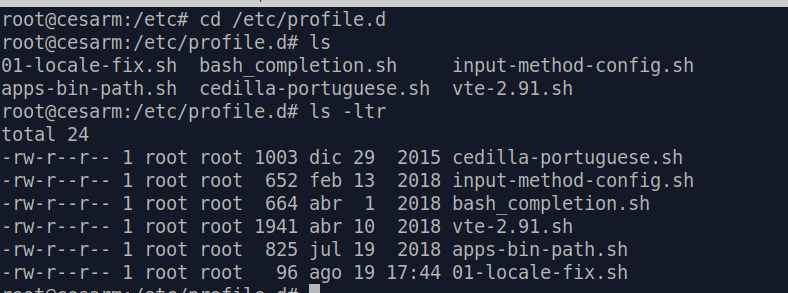


Accedamos el terminal numero 2 (tty2) como usuario raíz



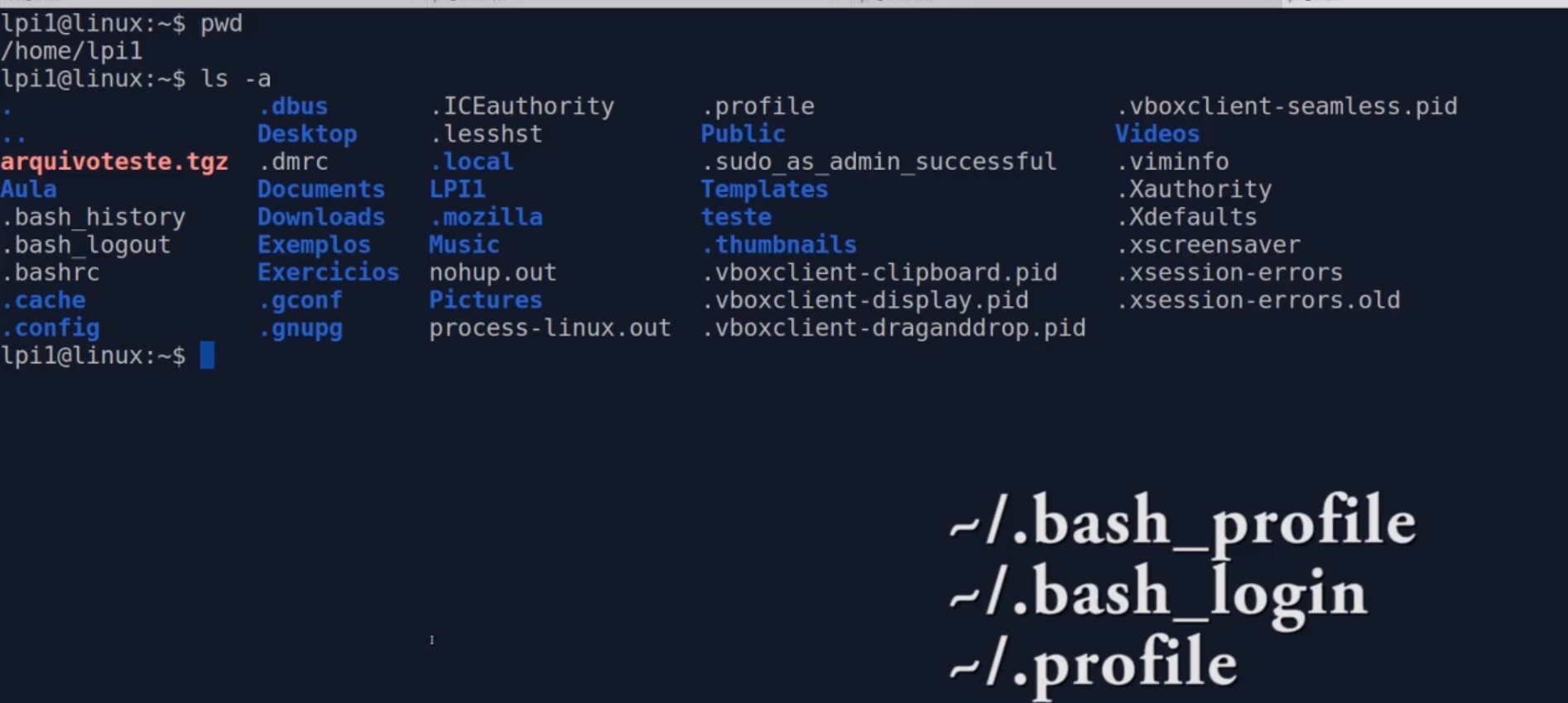


Entonces estas son las diferencias entre el etc/bash.bashrc y el etc/profile



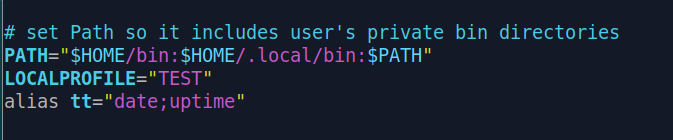
Cada vez que nos logueamos; estos scripts se van a cargar. Entonces son estos 2 archivos que hacen las definiciones generales que son aplicadas a todos los usuarios.

Dependiendo de la distribución de Linux podríamos tener algunos de los archivos que se muestran en la imagen a continuación:

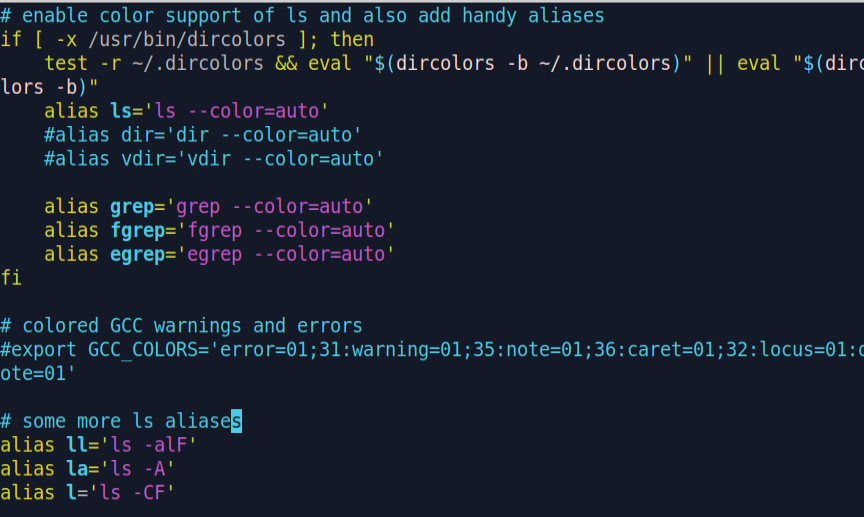


**.profile:** Es donde voy a colocar mis definiciones personales que van a ser colocadas en todo login nuevo.

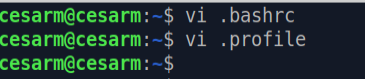
**vi .profile:** vamos a realizar los siguiente cambios al final del documento, para que cada vez que nos logueamos cargue estas variables.



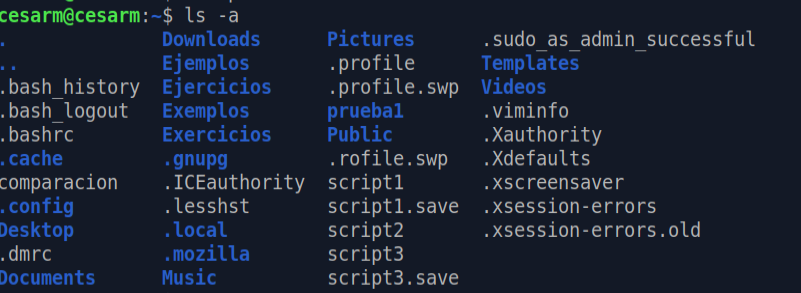
**.bashrc:** Este si es correspondiente a toda vez que realizamos una nueva sesión. Aquí tenemos por ejemplo varios aliases que están definidos aquí:



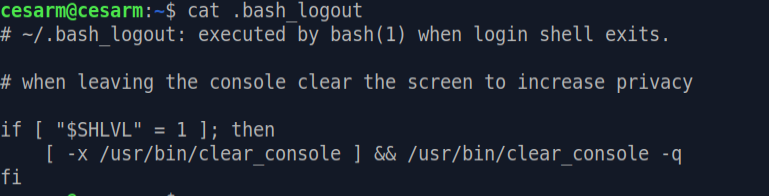
Entonces estos son los archivos utilizados para hacer definiciones globales de todos los usuarios y de cada usuario específicamente.



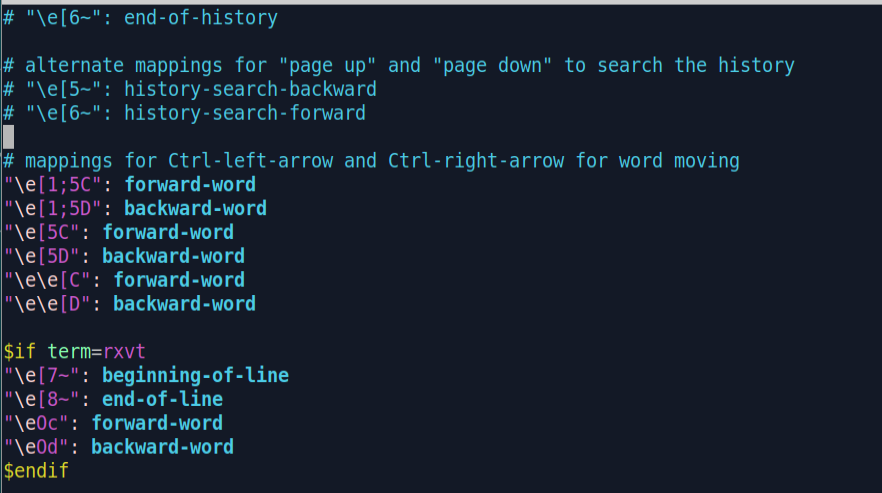
El usuario tiene otro archivo importante que es el .bash\_logout



Esto quiere decir lo que tenemos dentro de este archivo va a ser ejecutado toda vez que el usuario va a ser un logout:



**/etc/inputrc:** Es una forma de definir comandos en nuestro terminal, entonces si tenemos alguna pregunta de este tema se refiere a definiciones sobre el terminal.



**vi /etc/skel:** otra definición de ambiente importante es el directorio esqueleto los archivos que estén dentro van a ser creados cada vez que hagamos un nuevo usuario en el home del usuario.

