

Universidad Tecnológica de Panamá
Sistemas Operativos I
Experiencia Práctica en Laboratorio No. 8
El Shell de Linux

Prof. Aris Castillo de Valencia

Anel Atencio – 8-950-868

Objetivos:

1. Utilizar la línea de comandos para desarrollar tareas básicas
2. Usar y modificar el ambiente Shell incluido y definir, referenciar y exportar variables
3. Invocar comandos dentro y fuera de la ruta definida.

Metas:

Con esta experiencia práctica se espera que el estudiante sea capaz de realizar tareas relacionadas con el Shell del sistema operativo Linux/GNU a través de comandos.

Contenidos:

- Shells de Linux
- Variables de entorno

Metodología:

Se basa en métodos intuitivos, de experimentación y demostración en que se acerca al estudiante a situaciones reales de la práctica profesional de manera que resuelva las situaciones presentadas.

Evaluación:

- Se dará 50 puntos por el desarrollo de la práctica en el aula.
- Se dará 50 puntos por la entrega del informe escrito debidamente completado y por su nivel técnico.

Recursos:

- Hardware: computadora, conexión a Internet.
- Software: Sistema operativo Linux/GNU.

Procedimiento:

- Lea cuidadosamente la guía; pruebe cada uno de los comandos listados prestando especial atención a los resultados obtenidos y a las variantes que le ofrecen las opciones de los comandos. Ponga en práctica los comandos aprendidos haciendo los ejercicios sugeridos. Desarrolle la retroinformación y súbala a la plataforma Moodle.

Introducción

Shell o intérprete de comandos de GNU/Linux: es el medio que tenemos para interactuar con la máquina con la que estamos trabajando y con su sistema operativo. Una gran parte del trabajo de UNIX consiste en emitir órdenes. Emitir una orden significa que nos estamos comunicando con el SHELL.



Bash: Existen muchos tipos de shell, el más popular es 'bash' (bourne again shell). **Bash** es un programa informático basado en Shell cuya función consiste en interpretar órdenes.

La shell **bash** se puede utilizar de modo interactivo o como un lenguaje de escritura de gran alcance. Tras el arranque, **bash** ejecuta los comandos hallados en el archivo ~/.bashrc, permitiéndoles a los usuarios personalizar su shell. La shell **bash** guarda el historial de las líneas de comando ejecutadas.

Ejemplo: Con frecuencia, los programadores de lenguajes compilados tales como C suelen hallarse en un ciclo repetitivo: editar un archivo, compilarlo y luego ejecutar el programa. A continuación, un usuario edita un archivo que contiene un programa pequeño C y luego lo compila con el compilador C **gcc**. Después de ejecutar el programa, decide hacer algunos cambios. Hace entonces uso del historial de comandos para agilizar el proceso.

```
root@ubuntu:~# vi hello.c
root@ubuntu:~# gcc -o hello hello.c
root@ubuntu:~# ./hello
hello world
root@ubuntu:~# !n
nano hello.c
```

```
root@ubuntu:~# !g
gcc -o hello hello.c
root@ubuntu:~# !.
./hello
hello dolly
root@ubuntu:~# █
```

Nota: Observe que la shell **bash** imprime el comando seleccionado desde el historial del usuario antes de ejecutar el comando.

```
near@anel-VirtualBox:~/Documentos/lab8$ vi hello.c
near@anel-VirtualBox:~/Documentos/lab8$ cat hello.c
#include <stdio.h>
int main()
{
    printf("Hola mundo.\n");
    return 0;
}
near@anel-VirtualBox:~/Documentos/lab8$ gcc -o hello hello.c
near@anel-VirtualBox:~/Documentos/lab8$ ./hello
Hola mundo.
near@anel-VirtualBox:~/Documentos/lab8$ !n
bash: !n: event not found
near@anel-VirtualBox:~/Documentos/lab8$ !g
gcc -o hello hello.c
near@anel-VirtualBox:~/Documentos/lab8$ !.
./hello
Hola mundo.
near@anel-VirtualBox:~/Documentos/lab8$ █
```

Ordenes más comunes:

Comando pwd:

Variable que muestra la ruta del directorio actual. Esto es muy útil, cuando estamos dentro de muchos directorios y no sabemos dónde nos encontramos.

Sintaxis: **pwd**

```
root@ubuntu:~# pwd
/home/darlene
root@ubuntu:~# █
```

```
near@anel-VirtualBox:~/Documentos/lab8$ pwd
/home/near/Documentos/lab8
near@anel-VirtualBox:~/Documentos/lab8$
```

Comando uname: este comando te devuelve información sobre el sistema que estas usando.

Sintaxis: **uname** [opción]

Opciones:

- **-a:** da toda la información.
- **-r:** versión del kernel

```
root@ubuntu:~# uname -a
Linux ubuntu 2.6.38-8-generic #42-Ubuntu SMP Mon Apr 11 03:31:50 UTC 2011 i686 i
686 i386 GNU/Linux
root@ubuntu:~# uname -v
#42-Ubuntu SMP Mon Apr 11 03:31:50 UTC 2011
```

```
near@anel-VirtualBox:~/Documentos/lab8$ uname -a
Linux anel-VirtualBox 5.4.0-48-generic #52-Ubuntu SMP Thu Sep 10 10:58:49 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
near@anel-VirtualBox:~/Documentos/lab8$ uname -v
#52-Ubuntu SMP Thu Sep 10 10:58:49 UTC 2020
near@anel-VirtualBox:~/Documentos/lab8$ uname -r
5.4.0-48-generic
near@anel-VirtualBox:~/Documentos/lab8$
```

Comando history:

Este comando almacena en el buffer de memoria los comandos que se han ido ejecutando en la shell y con él es posible conocer cuales se han efectuado, y citarlos por su número de orden.

Sintaxis: **history**

```
1497 clear
1498 sudo -s
1499 gedit /etc/dtn.conf
1500 sudo -s
1501 ifconfig
1502 sudo -s
1503 dtntping localhost
1504 sudo -s
1505 gedit /etc/dtn.conf
1506 hostname ubuntu-a
1507 hostname
1508 dtntping localhost
1509 hostname
1510 hostname ubuntu-a
1511 sudo -s
1512 hostname ubuntu-a
1513 hostname
1514 sudo -s
1515 gedit /etc/dtn.conf
1516 dtntping localhost
1517 dtntping dtn://ubuntu.dtn
1518 dtntrecv dtn://ubuntu.dtn
1519 sudo -s
1520 dtntperf-server
```

```
near@anel-VirtualBox:~/Documentos/lab8$ history
1 cd
2 ls
3 cd Documentos
4 ls
5 mkdir lab8
6 cd lab8
7 ls
8 clear
9 vi hello.c
10 gcc -o hello hello.c
11 rm hello.c
12 ls
13 clear
14 vi hello.c
15 cat hello.c
16 gcc -o hello hello.c
17 rm hello.c
18 clear
19 vi hello.py
20 cat hello.py
21 gcc -o hello hello.py
22 rm hello.py
23 clear
24 ls
25 clear
26 vi hello.c
27 cat hello.c
28 gcc -o hello hello.c
29 rm hello.c
30 clear
31 vi hello.c
32 cat hello.c
33 gcc -o hello hello.c
34 rm hello.c
35 clear
36 vi hello.c
37 cat hello.c
38 gcc -o hello hello.c
39 rm hello.c
40 clear
41 vi hello.c
42 cat hello.c
43 gcc -o hello hello.c
44 clear
45 vi hello.c
46 rm hello.c
47 clear
48 vi hello.c
49 cat hello.c
50 gcc -o hello hello.c
```

La primera columna indica el número de comando, en el orden en el que se ha ejecutado. La segunda evidentemente es el comando que se ha ejecutado.

Si por ejemplo de la lista anterior se desea ejecutar un comando, se haría lo siguiente:

```
root@ubuntu:~# !1509
hostname
ubuntu
root@ubuntu:~#
```

```
67 gcc -o hello hello.c
68 ./hello
69 clear
70 pwd
71 uname -a
72 uname -v
73 uname -r
74 clear
75 history
near@anel-VirtualBox:~/Documentos/lab8$ !68
./hello
Hola mundo.
near@anel-VirtualBox:~/Documentos/lab8$
```

Comando echo:

Muestra en pantalla lo que se indica como argumento.

Sintaxis: **echo** *[argumento]*

```
root@ubuntu:~# echo casa
casa
```

```
near@anel-VirtualBox:~/Documentos/lab8$ echo Hola, me llamo Anel Atencio :D
Hola, me llamo Anel Atencio :D
near@anel-VirtualBox:~/Documentos/lab8$
```

Variables de Entorno

Las variables de entorno son porciones de memoria que el sistema toma para guardar valores específicos necesarios para el sistema.

Comando set

Nos sirve para saber las variables de entorno que hay en el sistema. También se utiliza para definir un valor a las variables de entorno del usuario.

Sintaxis: **set** | **more**

```
root@ubuntu:~# set | more
BASH=/bin/bash
BASHOPTS=checkwinsize:cmdhist:expand_aliases:extglob:extquote:force_ignore:hist
append:interactive_comments:progcomp:promptvars:sourcepath
BASH_ALIASES=()
BASH_ARGC=()
BASH_ARGV=()
BASH_CMDS=()
BASH_COMPLETION=/etc/bash_completion
BASH_COMPLETION_COMPAT_DIR=/etc/bash_completion.d
BASH_COMPLETION_DIR=/etc/bash_completion.d
BASH_LINENO=()
BASH_SOURCE=()
BASH_VERSINFO=([0]="4" [1]="2" [2]="8" [3]="1" [4]="release" [5]="i686-pc-linux-
gnu")
BASH_VERSION='4.2.8(1)-release'
```

```
near@anel-VirtualBox:~/Documentos/lab8$ set | more
BASH=/usr/bin/bash
BASHOPTS=checkwinsize:cmdhist:complete_fullquote:expand_aliases:extglob:extquote:force_ignore:globasciiranges:histappend:interactive_comments:progcomp:promptvars:sourcepath
BASH_ALIASES=()
BASH_ARGC=([0]="0")
BASH_ARGV=()
BASH_CMDS=()
BASH_COMPLETION_VERSINFO=([0]="2" [1]="10")
BASH_LINENO=()
BASH_SOURCE=()
BASH_VERSINFO=([0]="5" [1]="0" [2]="17" [3]="1" [4]="release" [5]="x86_64-pc-linux-gnu")
BASH_VERSION='5.0.17(1)-release'
CLUTTER_IM_MODULE=ibus
COLORTERM=truecolor
COLUMNS=177
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1001/bus
DESKTOP_SESSION=ubuntu
DIRSTACK=()
DISPLAY=:1
EUID=1001
GDMSESSION=ubuntu
GJS_DEBUG_OUTPUT=stderr
GJS_DEBUG_TOPICS='JS ERROR;JS LOG'
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GNOME_SHELL_SESSION_MODE=ubuntu
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/39b83did_a6fa_4fb4_bea7_e84f84974437
GNOME_TERMINAL_SERVICE=:1.174
GPG_AGENT_INFO=/run/user/1001/gnupg/S.gpg-agent:0:1
GROUPS=()
GTK_IM_MODULE=ibus
GTK_MODULES=gall:atk-bridge
HISTCONTROL=ignoreboth
HISTFILE=/home/near/.bash_history
```

Comando unset:

Remueve definiciones de variables o funciones.

Sintaxis: **unset** *[variable]*

Ejemplo:

```
root@ubuntu:~# echo $edad
23
root@ubuntu:~# unset edad
root@ubuntu:~# echo $edad

root@ubuntu:~#
```

```
near@anel-VirtualBox:~$ edad=20
near@anel-VirtualBox:~$ nombre=Anel
near@anel-VirtualBox:~$ echo $edad
20
near@anel-VirtualBox:~$ echo $nombre
Anel
near@anel-VirtualBox:~$
```

Comando env:

Muestra las variables de entorno definidas para el usuario actual.

Sintaxis: **env**

Ejemplo:

```
root@ubuntu:~# env
SHELL=/bin/bash
TERM=xterm
USER=root
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arj=01;31:*.taz=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lz=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.rar=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.axv=01;35:*.anx=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.
near@anel-VirtualBox:~$ env
SHELL=/bin/bash
SESSION_MANAGER=local/anel-VirtualBox:@/tmp/.ICE-unix/4552,unix/anel-VirtualBox:/tmp/.ICE-unix/4552
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GTK_IM_MODULE=ibus
LANGUAGE=es:en
QT4_IM_MODULE=ibus
LC_ADDRESS=es_PA.UTF-8
GNOME_SHELL_SESSION_MODE=ubuntu
LC_NAME=es_PA.UTF-8
SSH_AUTH_SOCK=/run/user/1001/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
LC_MONETARY=es_PA.UTF-8
SSH_AGENT_PID=4420
GTK_MODULES=gail:atk-bridge
PWD=/home/near
LOGNAME=near
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=x11
GPG_AGENT_INFO=/run/user/1001/gnupg/S.gpg-agent:0:1
```


Obtención del valor de una variable:

Para obtener el valor de una variable del SHELL debemos indicar el nombre de la variable y anteponer el símbolo \$. Cuando el SHELL encuentra una palabra que comienza por \$ supone que es una variable, y sustituye su valor por la parición de la variable.

Ejemplo:

```
root@ubuntu:~# echo $HOSTNAME
ubuntu
root@ubuntu:~#
```

```
near@anel-VirtualBox:~$ echo $HOSTNAME
anel-VirtualBox
near@anel-VirtualBox:~$
```

Definición de variables en el SHELL:

Para definir una variable, empleamos la siguiente sintaxis: ***nombre=valor***.

Ejemplo:

```
root@ubuntu:~# edad=23
root@ubuntu:~# nombre=darlene
root@ubuntu:~# echo $edad
23
root@ubuntu:~# echo $nombre
darlene
root@ubuntu:~#
```

```
near@anel-VirtualBox:~$ edad=20
near@anel-VirtualBox:~$ nombre=Anel
near@anel-VirtualBox:~$ echo $edad
20
near@anel-VirtualBox:~$ echo $nombre
Anel
near@anel-VirtualBox:~$
```

Nota: Se recomienda que las variables de entorno creadas, estén en minúsculas, de esta manera, no se borra nada indispensable del sistema. Las variables de entorno, cuando las crea un usuario, solo se ejecutan en la terminal donde se crean, para poder ejecutarlas en todas las terminales tendríamos que exportarlas.

Comando export:

Exporta el valor de la variable de entorno que se especifique.

Sintaxis: **export** *nom_var*.

Ejemplo:

```
[root@ubuntu:~]$ export SECONDS=0
[root@ubuntu:~]$ echo $SECONDS
0
```

La variable \$SECONDS cuenta la cantidad de segundos desde que entramos al sistema; con este mandato la seteamos a "0".

```
near@anel-VirtualBox:~$ export SECONDS=0
near@anel-VirtualBox:~$ echo $SECONDS
10
near@anel-VirtualBox:~$ echo $SECONDS
1090
near@anel-VirtualBox:~$
```

Comando exec: Reemplaza el intérprete de comandos con el programa definido.

Retroinformación.

1. Obtenga el valor de la variable PATH. Describa la salida.

```
near@anel-VirtualBox:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
near@anel-VirtualBox:~$
```

La salida que nos dio nos muestra que la variable PATH tiene como valor una dirección, aunque no la entiendo mucho.

2. Defina las siguientes variables m_bienvenida, m_atencion, y m_prohibicion para enviarle mensajes al usuario, de acuerdo con el título correspondiente a cada variable.

```
near@anel-VirtualBox:~$ m_bienvenida='Bienvenido Usuario'
near@anel-VirtualBox:~$ m_atencion='Atencion Usuario'
near@anel-VirtualBox:~$ m_prohibicion='Esto está prohibido Usuario'
near@anel-VirtualBox:~$ echo $m_bienvenida
Bienvenido Usuario
near@anel-VirtualBox:~$ echo $m_atencion
Atencion Usuario
near@anel-VirtualBox:~$ echo $m_prohibicion
Esto está prohibido Usuario
near@anel-VirtualBox:~$
```

3. Utilice la orden **env** para verificar que las variables hayan sido creadas correctamente.

```

near@anel-VirtualBox:~$ env
SHELL=/bin/bash
SESSION_MANAGER=local/anel-VirtualBox:@/tmp/.ICE-unix/4552,unix/anel-VirtualBox:/tmp/.ICE-unix/4552
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GTK_IM_MODULE=ibus
LANGUAGE=es:en
QT4_IM_MODULE=ibus
LC_ADDRESS=es_PA.UTF-8
LC_GNOME_SHELL_SESSION_MODE=ubuntu
LC_NAME=es_PA.UTF-8
SSH_AUTH_SOCK=/run/user/1001/keyring/ssh
XMODIFIERS=@Xlib=ibus
DESKTOP_SESSION=ubuntu
LC_MONETARY=es_PA.UTF-8
SSH_AGENT_PID=4420
GTK_MODULES=gall:atk-bridge
PWD=/home/near
LOGNAME=near
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=x11
GPG_AGENT_INFO=/run/user/1001/gnupg/5.gpg-agent:0:1
XAUTHORITY=/run/user/1001/gdm/Xauthority
GJS_DEBUG_TOPICS=JS ERROR:JS LOG
WINDOWPATH=3
HOME=/home/near
USER=near
IM_CONFIG_PHASE=1
LC_PAPER=es_PA.UTF-8
LANG=es_ES.UTF-8
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33:cd=40;33:or=40;31:01:ml=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.lha=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.zlp=01;31:*.z=01;31:*.d2=01;31:*.gz=01;31:*.lzz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tze=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.wim=01;31:*.swm=01;31:*.dwm=01;31:*.e=01;31:31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pex=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.m4p=00;36:*.m4b=00;36:*.m4r=00;36:*.m4u=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:
XDG_CURRENT_DESKTOP=ubuntu:GNOME
VTE_VERSION=6003
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/26ff31bc_69b9_4357_a92a_1a8c8af50438
INVOCATION_ID=f69681d236c14753bbf43469362f0f05
MANAGERPID=4245
CLUTTER_IM_MODULE=ibus
GJS_DEBUG_OUTPUT=stderr
LESSCLOSE=/usr/bin/lesspipe %s %s
XDG_SESSION_CLASS=user
TERM=xterm-256color
LC_IDENTIFICATION=es_PA.UTF-8
LESSOPEN=| /usr/bin/lesspipe %s

```

```

GNOME_TERMINAL_SERVICE=:1.106
DISPLAY=:1
SHLVL=1
LC_TELEPHONE=es_PA.UTF-8
QT4_IM_MODULE=ibus
LC_MEASUREMENT=es_PA.UTF-8
XDG_RUNTIME_DIR=/run/user/1001
LC_TIME=es_PA.UTF-8
JOURNAL_STREAM=9:7A282
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/local/share:/usr/share:/var/lib/snapd/desktop
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
GNOME_SESSION=ubuntu
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1001/bus
LC_NUMERIC=es_PA.UTF-8
_=/usr/bin/env
near@anel-VirtualBox:~$

```

4. Ejecute las variables definidas.

```

near@anel-VirtualBox:~$ m_bienvenida='Bienvenido Usuario'
near@anel-VirtualBox:~$ m_atencion='Atencion Usuario'
near@anel-VirtualBox:~$ m_prohibicion='Esto está prohibido Usuario'
near@anel-VirtualBox:~$ echo $m_bienvenida
Bienvenido Usuario
near@anel-VirtualBox:~$ echo $m_atencion
Atencion Usuario
near@anel-VirtualBox:~$ echo $m_prohibicion
Esto está prohibido Usuario
near@anel-VirtualBox:~$

```

5. ¿Qué aprendió de esta experiencia? ¿Cómo considera que le puede ser útil?

De esta experiencia pude aprender que en la terminal se pueden crear variables, algo que no tenía ni idea, y también que aquí se pueden crear programas. Me pareció muy curioso que pudiera crear un programa del lenguaje C y ejecutarlo. Me gustaría intentar ver qué otros lenguajes puedo utilizar en la Terminal.

6. ¿Cómo considera que se puede mejorar esta experiencia? ¿Qué cambiaría?

Por ahora todo estuvo sencillo y bien explicado, por lo que no cambiaría mucho, solo un poco del comando "env" porque fue el único que no comprendí bien. Alguna explicación o muestra detallada de su función.

7. ¿Qué sugerencias puede aportar?

Por ahora solo que se explique un poco mejor el comando env. También la diferencia entre el comando echo y export para una variable. Porque pude ver que echo es para imprimir, pero tampoco entendí muy bien su diferencia con export [variable].

8. Incluya material de apoyo útil para compartir con su clase.

http://www.chuidiang.org/clinix/herramientas/basico/hola_mundo.php

Esta página me hizo recordar un poco como hacer un código en C y también a ver mejor los comandos para entender un poco más.

Bibliografía:

1. Kernighan, B. y Pike, R. El Entorno de programación Unix. Prentice Hall.
2. Husain, Kamran y Parker, Timoty, et al. **Linux Unleashed**. Second Edition.