

GUION 1

1. Entra al sistema y comprueba si te funcionan los distintos tipos de terminales. Deberás identificarte al menos en uno de ellos. Indica cuál es el gráfico.

La terminal grafica es la que abrimos desde la interfaz grafica con CTRL+ALT+T.

2. Vete a tu directorio de usuario, compruébalo con pwd . Crea un directorio a partir de tu directorio llamado temporal, entra en él, y compruébalo de nuevo con pwd . Intenta hacer lo mismo desde el directorio /remotehome ¿Puedes hacerlo? Di porqué. Averigua si cd es interno o externo. ¿Y pwd?

Creamos el directorio temporal con el comando mkdir en nuestro directorio de usuario, luego lo intentamos en el directorio /remotehome y no se puede ya que no disponemos de permisos para escribir en él.

Cd es un comando interno y pwd externo, porque está en /bin

3. Ve a ~/temporal y con el comando touch (si no sabes que es prueba el man) crea estos ficheros: “pepe1”, “pepe2”, “pepe3”, “pepe12” y “pipe3”. Copia, mueve y enlaza de dos formas (hardware y software) algún fichero de temporal en su directorio padre. Si haces el enlace blando de forma relativa ¿qué ocurre? ¿Cómo lo solucionarías? Explora las opciones de los comandos. Localiza en que sección están del manual.

Hacemos cd temporal y nos colocamos en este directorio y en él hace touch pepe1, touch pep2 ,etc para crear los ficheros.

Para crear el enlace hardware hacemos ln -P pepe1 .. y para el software ln -s /home/diego/temporal/pepe1 /home/diego/.

Cuando creamos el enlace blando de forma relativa se crea la “copia” pero el enlace esta roto. Para arreglarlo hay dos opciones: utilizar la ruta absoluta o trabajar en el directorio en el que creamos el enlace.

4. Haz un ls con diversas opciones (a, l, s, r, y R) y describe las diferencias. Cuando termines borra el directorio temporal con rmdir. Indica si puedes hacerlo o no. Si es que no, ¿Cómo lo harías (pero no lo hagas)?

ls -a Muestra todo

ls -l muestra archivos con sus permisos y última fecha de modificación

ls -s muestra lo que ocupa cada archivo en bloques

ls -r muestra ordenado alfabéticamente inverso

ls -R muestra el contenido de los directorios

No se pude hacer con rmdir solo ya que el directorio tiene contenido, para borrarlo usaríamos rm -rf.

5. Usa el comando history y el comodín "!". Prueba el ctrl+r (búsqueda reversa) y el resto de combinaciones de teclas para editar líneas. Comprueba como cuando se hace uso de los paréntesis, se ejecuta lo encerrado en una subshell. Prueba el concatenado condicional para copiar un fichero que exista o no, si es así, que se edite. Haz lo mismo compilando un programa con y sin errores y ejecutándolo.

Copia de fichero que si lo hace se abre: cp pepe 1 .. && gedit pepe1

Programa en C que si compila bien se ejecuta:

ej5.c:

```
#include <stdio.h>
```

```
int main(){
```

```
    printf("Hello world!");
```

```
    return 0;
```

```
}
```

```
gcc ej5.c && ./a.out
```

6.Vete a temporal y prueba con los ficheros creados los caracteres comodines ("*", "?", "[]"). Utiliza también el comodín de negación (es necesario ejecutar la tcsh). Crea un fichero con un espacio en su nombre y bórralo. Vete a tu directorio con "~". Usa el comando echo con un argumento que no quepa en una línea (uso de \).

ls p{ep,ip}* muestra los fichero que empiezan por p y luego siguen por ep o ip y sus siguientes valores pueden ser cualquiera.

mkdir diego\ hola crea el directorio con un espacio

rmdir diego\ hola borra ese directorio

echo cualquier\

cosa

7. Usando la redirección de salida crea un fichero calendario con el comando cal, primero de 2017 y después añadiendo el 2016. Mira el contenido del fichero con cat, con more y con less usando sus subcomandos (retorno, espacio, b, q, etc.).

```
cal 2017>calendario
cal 2016>>calendario
```

```
victor@victor-Lenovo-G50-80:~$ cal 2017 >calendario
victor@victor-Lenovo-G50-80:~$ cal 2016 >>calendario
victor@victor-Lenovo-G50-80:~$ cat calendario
2017
Enero          Febrero          Marzo
do lu ma mi ju vi sa do lu ma mi ju vi sa do lu ma mi ju vi sa
1 2 3 4 5 6 7          1 2 3 4          1 2 3 4
8 9 10 11 12 13 14      5 6 7 8 9 10 11      5 6 7 8 9 10 11
15 16 17 18 19 20 21    12 13 14 15 16 17 18    12 13 14 15 16 17 18
22 23 24 25 26 27 28    19 20 21 22 23 24 25    19 20 21 22 23 24 25
29 30 31              26 27 28              26 27 28 29 30 31

Abril          Mayo          Junio
do lu ma mi ju vi sa do lu ma mi ju vi sa do lu ma mi ju vi sa
1              1 2 3 4 5 6          1 2 3
2 3 4 5 6 7 8      7 8 9 10 11 12 13      4 5 6 7 8 9 10
9 10 11 12 13 14 15 14 15 16 17 18 19 20      11 12 13 14 15 16 17
16 17 18 19 20 21 22 21 22 23 24 25 26 27      18 19 20 21 22 23 24
23 24 25 26 27 28 29 28 29 30 31      25 26 27 28 29 30
30

Julio          Agosto          Septiembre
do lu ma mi ju vi sa do lu ma mi ju vi sa do lu ma mi ju vi sa
1              1 2 3 4 5          1 2
2 3 4 5 6 7 8      6 7 8 9 10 11 12      3 4 5 6 7 8 9
9 10 11 12 13 14 15 13 14 15 16 17 18 19      10 11 12 13 14 15 16
16 17 18 19 20 21 22 20 21 22 23 24 25 26      17 18 19 20 21 22 23
23 24 25 26 27 28 29 27 28 29 30 31      24 25 26 27 28 29 30
30 31

Octubre        Noviembre        Diciembre
do lu ma mi ju vi sa do lu ma mi ju vi sa do lu ma mi ju vi sa
1 2 3 4 5 6 7      1 2 3 4          1 2
8 9 10 11 12 13 14 5 6 7 8 9 10 11      3 4 5 6 7 8 9
15 16 17 18 19 20 21 12 13 14 15 16 17 18    10 11 12 13 14 15 16
22 23 24 25 26 27 28 19 20 21 22 23 24 25    17 18 19 20 21 22 23
29 30 31          26 27 28 29 30      24 25 26 27 28 29 30
31

2016
Enero          Febrero          Marzo
do lu ma mi ju vi sa do lu ma mi ju vi sa do lu ma mi ju vi sa
1 2          1 2 3 4 5 6          1 2 3 4 5
```

Con cat se nos muestra todo el fichero de texto cargado el terminal.

```
2017
Enero          Febrero          Marzo
do lu ma mi ju vi sa do lu ma mi ju vi sa do lu ma mi ju vi sa
1 2 3 4 5 6 7          1 2 3 4          1 2 3 4
8 9 10 11 12 13 14      5 6 7 8 9 10 11      5 6 7 8 9 10 11
15 16 17 18 19 20 21    12 13 14 15 16 17 18    12 13 14 15 16 17 18
22 23 24 25 26 27 28    19 20 21 22 23 24 25    19 20 21 22 23 24 25
29 30 31              26 27 28              26 27 28 29 30 31

Abril          Mayo          Junio
do lu ma mi ju vi sa do lu ma mi ju vi sa do lu ma mi ju vi sa
1              1 2 3 4 5 6          1 2 3
2 3 4 5 6 7 8      7 8 9 10 11 12 13      4 5 6 7 8 9 10
9 10 11 12 13 14 15 14 15 16 17 18 19 20      11 12 13 14 15 16 17
16 17 18 19 20 21 22 21 22 23 24 25 26 27      18 19 20 21 22 23 24
23 24 25 26 27 28 29 28 29 30 31      25 26 27 28 29 30
30

Julio          Agosto          Septiembre
do lu ma mi ju vi sa do lu ma mi ju vi sa do lu ma mi ju vi sa
1              1 2 3 4 5          1 2
2 3 4 5 6 7 8      6 7 8 9 10 11 12      3 4 5 6 7 8 9
9 10 11 12 13 14 15 13 14 15 16 17 18 19      10 11 12 13 14 15 16
16 17 18 19 20 21 22 20 21 22 23 24 25 26      17 18 19 20 21 22 23
23 24 25 26 27 28 29 27 28 29 30 31      24 25 26 27 28 29 30
30 31

Octubre        Noviembre        Diciembre
do lu ma mi ju vi sa do lu ma mi ju vi sa do lu ma mi ju vi sa
1 2 3 4 5 6 7      1 2 3 4          1 2
8 9 10 11 12 13 14 5 6 7 8 9 10 11      3 4 5 6 7 8 9
15 16 17 18 19 20 21 12 13 14 15 16 17 18    10 11 12 13 14 15 16
22 23 24 25 26 27 28 19 20 21 22 23 24 25    17 18 19 20 21 22 23
29 30 31          26 27 28 29 30      24 25 26 27 28 29 30
31

2016
Enero          Febrero          Marzo
do lu ma mi ju vi sa do lu ma mi ju vi sa do lu ma mi ju vi sa
1 2          1 2 3 4 5 6          1 2 3 4 5
3 4 5 6 7 8 9      7 8 9 10 11 12 13      6 7 8 9 10 11 12
10 11 12 13 14 15 16 14 15 16 17 18 19 20    13 14 15 16 17 18 19
--Más-- (58%)
```

Al hacer more se nos carga parte del fichero y luego pulsando intro se va cargando parte hasta el final

							2017														
Enero							Febrero							Marzo							
do	lu	ma	mi	ju	vi	sá	do	lu	ma	mi	ju	vi	sá	do	lu	ma	mi	ju	vi	sá	
1	2	3	4	5	6	7					1	2	3	4				1	2	3	4
8	9	10	11	12	13	14	5	6	7	8	9	10	11	5	6	7	8	9	10	11	
15	16	17	18	19	20	21	12	13	14	15	16	17	18	12	13	14	15	16	17	18	
22	23	24	25	26	27	28	19	20	21	22	23	24	25	19	20	21	22	23	24	25	
29	30	31					26	27	28					26	27	28	29	30	31		
Abril							Mayo							Junio							
do	lu	ma	mi	ju	vi	sá	do	lu	ma	mi	ju	vi	sá	do	lu	ma	mi	ju	vi	sá	
						1					1	2	3	4				1	2	3	
2	3	4	5	6	7	8	7	8	9	10	11	12	13	4	5	6	7	8	9	10	
9	10	11	12	13	14	15	14	15	16	17	18	19	20	11	12	13	14	15	16	17	
16	17	18	19	20	21	22	21	22	23	24	25	26	27	18	19	20	21	22	23	24	
23	24	25	26	27	28	29	28	29	30	31				25	26	27	28	29	30		
30																					
Julio							Agosto							Septiembre							
do	lu	ma	mi	ju	vi	sá	do	lu	ma	mi	ju	vi	sá	do	lu	ma	mi	ju	vi	sá	
						1					1	2	3	4					1	2	
2	3	4	5	6	7	8	6	7	8	9	10	11	12	3	4	5	6	7	8	9	
9	10	11	12	13	14	15	13	14	15	16	17	18	19	10	11	12	13	14	15	16	
16	17	18	19	20	21	22	20	21	22	23	24	25	26	17	18	19	20	21	22	23	
23	24	25	26	27	28	29	27	28	29	30	31			24	25	26	27	28	29	30	
30	31																				
Octubre							Noviembre							Diciembre							
do	lu	ma	mi	ju	vi	sá	do	lu	ma	mi	ju	vi	sá	do	lu	ma	mi	ju	vi	sá	
1	2	3	4	5	6	7					1	2	3	4					1	2	
8	9	10	11	12	13	14	5	6	7	8	9	10	11	3	4	5	6	7	8	9	
15	16	17	18	19	20	21	12	13	14	15	16	17	18	10	11	12	13	14	15	16	
22	23	24	25	26	27	28	19	20	21	22	23	24	25	17	18	19	20	21	22	23	
29	30	31					26	27	28	29	30			24	25	26	27	28	29	30	
														31							
Enero							2016 Febrero							Marzo							
do	lu	ma	mi	ju	vi	sá	do	lu	ma	mi	ju	vi	sá	do	lu	ma	mi	ju	vi	sá	
						1					1	2	3	4				1	2	3	4
3	4	5	6	7	8	9	7	8	9	10	11	12	13	6	7	8	9	10	11	12	
10	11	12	13	14	15	16	14	15	16	17	18	19	20	13	14	15	16	17	18	19	
calendario																					

10. Crea un fichero con el comando cal. Busca dentro del fichero creado con el comando grep las ocurrencias de la palabra febrero.

cal 1990 > calendario.txt; grep Febrero calendario.txt

```
diego@diego-VirtualBox:~/desarrollo$ cal 1990 > calendario.txt|grep Febrero calendario.txt
Enero      Febrero    Marzo
diego@diego-VirtualBox:~/desarrollo$ cat calendario.txt
1990
Enero      Febrero    Marzo
do lu ma mi ju vi sa do lu ma mi ju vi sa do lu ma mi ju vi sa
1 2 3 4 5 6          1 2 3          1 2 3
7 8 9 10 11 12 13    4 5 6 7 8 9 10    4 5 6 7 8 9 10
14 15 16 17 18 19 20 11 12 13 14 15 16 17 11 12 13 14 15 16 17
21 22 23 24 25 26 27 18 19 20 21 22 23 24 18 19 20 21 22 23 24
28 29 30 31          25 26 27 28          25 26 27 28 29 30 31

Abril      Mayo       Junio
do lu ma mi ju vi sa do lu ma mi ju vi sa do lu ma mi ju vi sa
1 2 3 4 5 6 7        1 2 3 4 5          1 2
8 9 10 11 12 13 14    6 7 8 9 10 11 12    3 4 5 6 7 8 9
15 16 17 18 19 20 21 13 14 15 16 17 18 19 10 11 12 13 14 15 16
22 23 24 25 26 27 28 20 21 22 23 24 25 26 17 18 19 20 21 22 23
29 30                27 28 29 30 31    24 25 26 27 28 29 30

Julio      Agosto     Septiembre
do lu ma mi ju vi sa do lu ma mi ju vi sa do lu ma mi ju vi sa
1 2 3 4 5 6 7        1 2 3 4          1
8 9 10 11 12 13 14    5 6 7 8 9 10 11    2 3 4 5 6 7 8
15 16 17 18 19 20 21 12 13 14 15 16 17 18 9 10 11 12 13 14 15
22 23 24 25 26 27 28 19 20 21 22 23 24 25 16 17 18 19 20 21 22
29 30 31            26 27 28 29 30 31    23 24 25 26 27 28 29
30

Octubre    Noviembre   Diciembre
do lu ma mi ju vi sa do lu ma mi ju vi sa do lu ma mi ju vi sa
1 2 3 4 5 6          1 2 3          1
7 8 9 10 11 12 13    4 5 6 7 8 9 10    2 3 4 5 6 7 8
14 15 16 17 18 19 20 11 12 13 14 15 16 17 9 10 11 12 13 14 15
21 22 23 24 25 26 27 18 19 20 21 22 23 24 16 17 18 19 20 21 22
28 29 30 31          25 26 27 28 29 30    23 24 25 26 27 28 29
30 31
```

11. Realizar un pipeline (tubería) con el comando ls (en forma recursiva del directorio /) y el comando grep para buscar los ficheros que tengan “pepe” en su nombre y acaben en un dígito.

(sudo ls / -R | egrep pepe[[:digit:]]\$)

```
diego@diego-VirtualBox:~$ (sudo ls / -R | egrep pepe[[:digit:]]$)
pepe1
pepe1
pepe2
pepe3
ls: cannot open directory '/run/user/1000/gvfs': Permission denied
diego@diego-VirtualBox:~$
```

12. Averigua los cinco ficheros más grandes de todo tu directorio de dos maneras diferentes: usando find y usando directamente ls. Pista: deberás usar los comandos sort y head canalizados.

```
sudo find / -type s -printf '%d %p\n'|sort -nr|head -5
```

Busca todos los 5 ficheros más grandes ordenados de menor a mayor mostrando su tamaño y su nombre

```
ls -s | sort -nr | head -5
```

13. Indica que dos procesos están siempre cuando se ejecuta ps. Averigua las opciones del comando ps. Ejecuta en *background* el editor de textos vi. Mira si el proceso se ha creado realmente. Si es así mátalo (kill). Repítelo con ejecución en *foreground* y para al proceso (debes saber cómo hacerlo). Después, pásalo de nuevo a *foreground*, páralo de nuevo, pásalo a *background* (indica la diferencia entre este estado y el anterior) y finalmente mátalo con killall. Piensa en una forma de averiguar la shell que estás ejecutando.

Siempre se están ejecutando .bash y ps.

kill -9 6025 (en ese instante el PID era 6025) Así matamos el proceso de vi creado en background.

vi->ctrl z->fg 1->bg 1->killall -9 vi Se crea el proceso de vi , salimos de el y lo poner en foreground, lo volvemos a pasar a background y matamos todos los procesos vi.

La diferencia entre foreground y background es que en fg ves el programa y puedes utilizarlo mientras que en bg el proceso sigue ejecutándose pero no puedes hacer nada con el excepto pasarlo a fg o matarlo

Con ps ya veo que estoy utilizando bash.

14. Si estás ejecutando una bash, mira el valor de las variables de shell PS1 y PATH (entorno) de dos formas; utilizando echo y una canalización con el grep. Arranca una shell de tipo tcsh (si está instalada) y mira el valor esta vez, si están indefinidas piensa el porqué. Sal de la tcsh. Mira todas las variables que tienes definidas, averigua la shell que estás usando y el tipo de terminal. Cambia el valor de PS1 para que te ponga un mensaje personal terminado en \$.

A terminal window titled 'victor@victor-Lenovo-G50-80: ~' with a dark purple background. The user runs 'echo \$PS1' and 'echo \$PATH'. The output for PS1 is a complex escape sequence. The output for PATH lists various system and user directories. Then, the user runs 'echo \$PS1|grep '' and 'echo \$PATH|grep '' to filter the output. The terminal shows the following commands and their outputs:

```
victor@victor-Lenovo-G50-80:~$ echo $PS1
\[ \e]0;\u@\h: \w\a\]${debian_chroot:+($debian_chroot)}\[ \033[01;32m\]\u@\h\[ \033[00m\]:\[ \033[01;34m\]\w\[ \033[00m\]\$
victor@victor-Lenovo-G50-80:~$ echo $PATH
/home/victor/bin:/home/victor/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/home/victor/bin:/usr/local/java/jdk1.8.0_20/bin:/usr/share/maven/bin:/usr/lib/jvm/java-8-openjdk-amd64/bin
victor@victor-Lenovo-G50-80:~$ echo $PS1|grep ''
\[ \e]0;\u@\h: \w\a\]${debian_chroot:+($debian_chroot)}\[ \033[01;32m\]\u@\h\[ \033[00m\]:\[ \033[01;34m\]\w\[ \033[00m\]\$
victor@victor-Lenovo-G50-80:~$ echo $PATH|grep ''
/home/victor/bin:/home/victor/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/home/victor/bin:/usr/local/java/jdk1.8.0_20/bin:/usr/share/maven/bin:/usr/lib/jvm/java-8-openjdk-amd64/bin
victor@victor-Lenovo-G50-80:~$
```

En el tcsh no está definida la variable PS1 porque pertenece a BASH y PATH si que esta porque es común para todas las shells.

printenv para mostrar todas las variables definidas.

```
victor@victor-Lenovo-G50-80:~$ printenv
XDG_VTNR=7
XDG_SESSION_ID=c2
CLUTTER_IM_MODULE=xim
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/victor
GPG_AGENT_INFO=/home/victor/.gnupg/S.gpg-agent:0:1
SHELL=/bin/bash
TERM=xterm-256color
VTE_VERSION=4205
QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1
WINDOWID=62914570
UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1448
GNOME_KEYRING_CONTROL=
GTK_MODULES=gail:atk-bridge:unity-gtk-module
USER=victor
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:mi=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:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=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:*.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:*.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:*.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:*.mid=00;36:*.midi=00;36:*.mka=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:
QT_ACCESSIBILITY=1
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/usr/share/upstart/xdg:/etc/xdg
PATH=/home/victor/bin:/home/victor/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/home/victor/bin:/usr/local/java/jdk1.8.0_20/bin:/usr/share/maven/bin:/usr/lib/jvm/java-8-openjdk-amd64/bin
DESKTOP_SESSION=ubuntu
QT_IM_MODULE=ibus
QT_QPA_PLATFORMTHEME=appmenu-qt5
XDG_SESSION_TYPE=x11
PWD=/home/victor
JOB=gnome-session
XMODIFIERS=@im=ibus
JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
GNOME_KEYRING_PID=
LANG=es_ES.UTF-8
PATH=/home/victor/bin:/home/victor/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/home/victor/bin:/usr/local/java/jdk1.8.0_20/bin:/usr/share/maven/bin:/usr/lib/jvm/java-8-openjdk-amd64/bin
DESKTOP_SESSION=ubuntu
QT_IM_MODULE=ibus
QT_QPA_PLATFORMTHEME=appmenu-qt5
XDG_SESSION_TYPE=x11
PWD=/home/victor
JOB=gnome-session
XMODIFIERS=@im=ibus
JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
GNOME_KEYRING_PID=
LANG=es_ES.UTF-8
GDM_LANG=es
MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path
IM_CONFIG_PHASE=1
COMPIZ_CONFIG_PROFILE=ubuntu
GDMSESSION=ubuntu
SESSIONTYPE=gnome-session
GTK2_MODULES=overlay-scrollbar
M2_HOME=/usr/share/maven
HOME=/home/victor
XDG_SEAT=seat0
SHLVL=1
LANGUAGE=es
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
UPSTART_INSTANCE=
UPSTART_EVENTS=started starting
XDG_SESSION_DESKTOP=ubuntu
LOGNAME=victor
QT4_IM_MODULE=xim
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/share/gnome:/usr/local/share:/usr/share:/var/lib/napd/desktop:/var/lib/napd/desktop
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-H253PmuKfJ
LESSOPEN=| /usr/bin/lesspipe %s
INSTANCE=Unity
UPSTART_JOB=unity-settings-daemon
XDG_RUNTIME_DIR=/run/user/1000
DISPLAY=:0
XDG_CURRENT_DESKTOP=Unity
GTK_IM_MODULE=ibus
LESSCLOSE=/usr/bin/lesspipe %s %s
XAUTHORITY=/home/victor/.Xauthority
_=/usr/bin/printenv
victor@victor-Lenovo-G50-80:~$ █
```

Todas la variables definidas.

Para averiguar la shell uso ps para ver cual se esta ejecutando y veo que es BASH
La terminal que estoy usando la veo con el comando tty

PS1="Lo que tu quieras \$ " ↵ cambias el mensaje personal al que tu quieras.

```
victor@victor-Lenovo-G50-80:~$ PS1=hola$
hola$cd
hola$ls
calendario Descargas eclipse Escritorio jdk1.8.0_151 Plantillas README.md Videos
calendario2017 Documentos eclipse-installer fichero.txt memoriasoftware.odt practica2 snap
calendario.txt Dropbox eclipse-workspace Imágenes Música
hola$
```

Como se ve cambia el mensaje personal.

15. Crea una variable de shell para asignar el nombre de un directorio a esa variable, comprueba su contenido y después úsala con el comando cd. Comprueba con grep si existe como variable de entorno, conviértela a entorno y comprueba de nuevo. Por último, elimínala.

export para crearla después de haberla metido en .bashrc y unset para eliminarla.

HOLA="/home/diego/temporal" en .bashrc, source para ejecutar .bashrc, export \$HOLA y compruebo con cd que funciona.

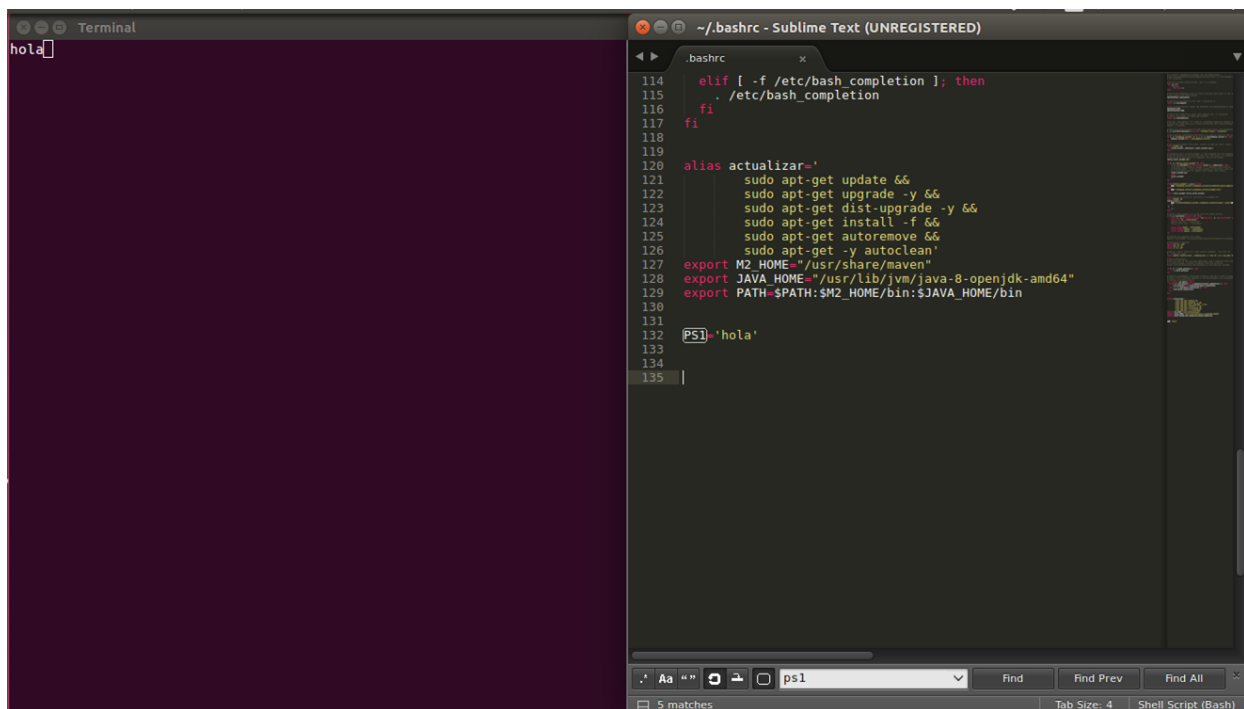
16. Averigua los ficheros de configuración que tienes ocultos en tu directorio personal. Examina el aspecto del que tienes que modificar para crear variables. Modifícalo (al final) para cambiar la variable adecuada para redefinir tu inductor (*prompt*).

ls -a para ver todos los ficheros, también los ocultos.

```
victor@victor-Lenovo-G50-80:~$ ls -a
.          .compiz      eclipse      .gnome      .mozilla    practica2    .steampid
..         .config      .eclipse    .gnupg      Música      .profile     .sudo_as_admin_successful
.bash_history .dbus        eclipse-installer .gphoto     .nano       Público      .swt
.bash_logout .Descargas   eclipse-workspace .ICAuthority .oracle_jre_usage .pulse-cookie temporal
.bashrc      .dmrc        Escritorio   Inágenes    .p2         README.md    .tooling
.cache       .Documentos  fichero.txt  .java       .pk1        snap         Videos
calendario   .dropbox     .gconf       jdk1.8.0_151 .Plantillas .sonar       .Xauthority
calendario2017 .Dropbox     .git         .local      .pract1-0c.swo .steam       .xsession-errors
calendario.txt .dropbox-dist .gitconfig   memoriasoftware.odt .pract1-0c.swp .steampath   .xsession-errors.old
victor@victor-Lenovo-G50-80:~$
```

Todos los ficheros del directorio incluyendo ocultos.

El fichero de configuración oculto es el .bashrc. Al final de este fichero ponemos PS1='lo que queramos', guardamos y volvemos a abrir la terminal viendo que el prompt ha cambiado.



```
hola
```

```
114 elif [ -f /etc/bash_completion ]; then
115     . /etc/bash_completion
116 fi
117
118
119
120 alias actualizar='
121     sudo apt-get update &&
122     sudo apt-get upgrade -y &&
123     sudo apt-get dist-upgrade -y &&
124     sudo apt-get install -f &&
125     sudo apt-get autoremove &&
126     sudo apt-get -y autoclean'
127 export M2_HOME="/usr/share/maven"
128 export JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64"
129 export PATH=$PATH:$M2_HOME/bin:$JAVA_HOME/bin
130
131
132 PS1='hola'
133
134
135 |
```

En el .bashrc al final definimos el PS1, volvemos a abrir la terminal y vemos que el prompt ha cambiado.

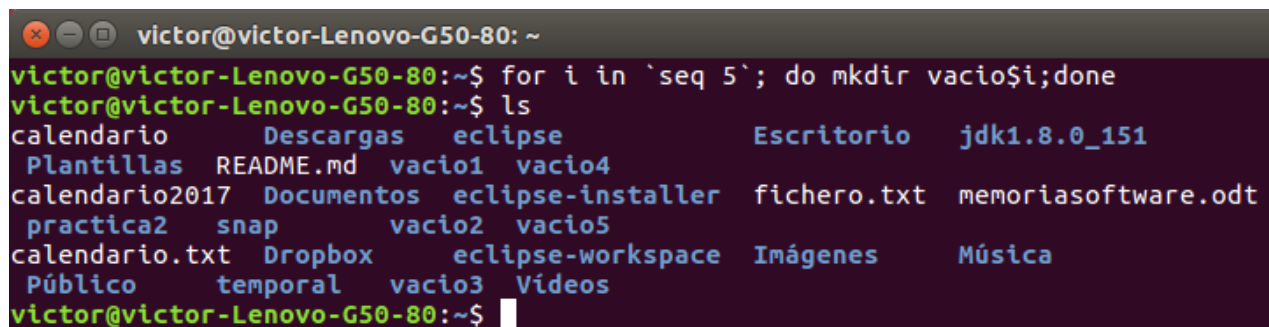
17. Vete a los ficheros de configuración de la Shell y comprueba si hay alias definidos. Si no es así crea un alias que se llame ll, si existe crea uno que te parezca útil.

```
alias actualizar='
    sudo apt-get update &&
    sudo apt-get upgrade -y &&
    sudo apt-get dist-upgrade -y &&
    sudo apt-get install -f &&
    sudo apt-get autoremove &&
    sudo apt-get -y autoclean'
```

Como ya está definido el alias ll, creamos un alias llamado actualizar que sirve para usar todos los comandos de actualizar y limpieza de una vez solo, sin tener que ir introduciéndolos de uno en uno.

18. Las macros también funcionan igual en línea de comando, separando cada línea por “;”. Crea con una única línea (y un bucle for) cinco ficheros vacíos que vayan de vacio1 a vacio5.

```
for i in `seq 5`; do mkdir vacio$i;done
```

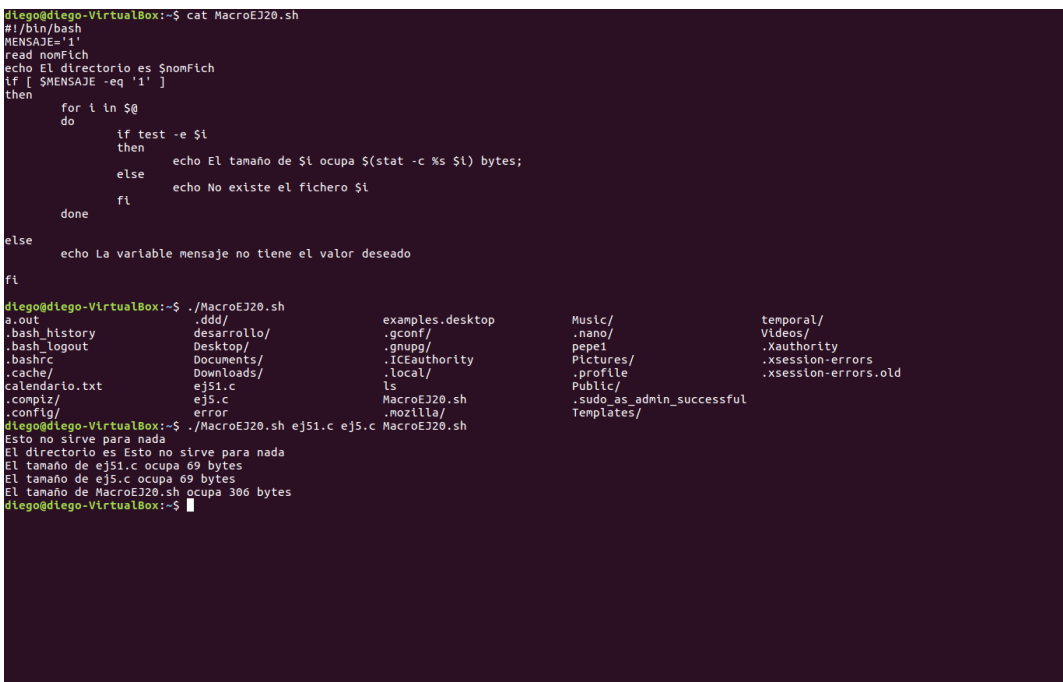


```
victor@victor-Lenovo-G50-80: ~
victor@victor-Lenovo-G50-80:~$ for i in `seq 5`; do mkdir vacio$i;done
victor@victor-Lenovo-G50-80:~$ ls
calendario      Descargas      eclipse          Escritorio      jdk1.8.0_151
Plantillas      README.md      vacio1          vacio4
calendario2017  Documentos     eclipse-installer fichero.txt      memoriasoftware.odt
practica2       snap          vacio2          vacio5
calendario.txt  Dropbox       eclipse-workspace Imágenes        Música
Público         temporal      vacio3          Videos
victor@victor-Lenovo-G50-80:~$
```

19. Edita con vi un fichero de texto. vi o vim es un editor de texto de pantalla básico, tienes un pequeño manual en el apartado posterior. El uso de vi a pesar de ser un editor primario, está justificado por su carácter universal y no depender de las características de un teclado particular. Al menos una vez en la vida es necesario su uso. Incluso, si te acostumbras a él, será más eficiente que cualquier otro editor gráfico).



20. Ese fichero será una macro que servirá para saber el tamaño en Kb o Mb de una serie de ficheros de un directorio. La macro tomará estos ficheros por argumento y pedirá el directorio con lectura. Sólo mostrará el tamaño de los ficheros si existen y si está definida una variable MENSAJE puesta a SI. La variable MENSAJE en un caso estará definida desde fuera de la macro y en el otro caso de forma interna ¿puedes acceder a la variable interna definida en la macro desde la shell?



En la imagen se muestra el script creado y su ejecución con 3 parámetros.

No se ha conseguido ejecutar correctamente la macro con la variable MENSAJE definida fuera de la macro.

21. Elige un fichero (o créalo) y cámbiale los permisos de lectura y escritura para que nadie pueda acceder a él. Intenta mostrar su contenido por pantalla, ¿qué ocurre?

```
diego@diego-VirtualBox:~$ ls
a.out      desarrollo  Documents  ej51.c  error      ls      Music  Pictures  Templates  Videos
calendario.txt  Desktop    Downloads  ej5.c   examples.desktop  MacroEJ20.sh  pepe1  Public    temporal
diego@diego-VirtualBox:~$ chmod -rw MacroEJ20.sh
diego@diego-VirtualBox:~$ cat MacroEJ20.sh
cat: MacroEJ20.sh: Permission denied
diego@diego-VirtualBox:~$
```

Como se puede ver en la imagen, después de quitarle los permisos no podemos ni leer ni escribir en él.

22. Crea dos terminales sobre ventanas o usa alguno de texto. Comprueba que estás en el sistema con who (who am i). Utiliza el comando write contigo mismo. Prueba mesg. Con el comando ps mira el dispositivo que estás usando en la shell. Observa el contenido del directorio /dev hasta encontrarlo. Usa la redirección para mostrar con cat el contenido de un fichero en el dispositivo que usas (puede ser tty o pts/x) de forma equivalente a como lo harías con un fichero. Hazlo también para que el contenido salga en el otro terminal.

Para enviarme mensajes a mi mismo abro dos terminales y me logeo en las 2, en ambas me doy para que me escriban. Escribo write diego en las 2 terminales y ya puedes comunicarte.

```
diego@diego-VirtualBox:~$ sudo login diego
Password:
Last login: dom mar  4 14:41:56 CET 2018 on pts/17
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.10.0-28-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

374 packages can be updated.
221 updates are security updates.

diego@diego-VirtualBox:~$ mesg y
diego@diego-VirtualBox:~$ tty
/dev/pts/4
diego@diego-VirtualBox:~$ write diego 17
write: diego is not logged in on 17
diego@diego-VirtualBox:~$ write diego
write: diego is logged in more than once; writing to pts/17
Message from diego@diego-VirtualBox on pts/17 at 14:43 ...
Esto lo escribo desde la terminal de la izquierda
Esto lo escribo desde la terminal de la derecha
^

diego@diego-VirtualBox:~$ sudo login diego
Password:
Last login: dom mar  4 14:42:37 CET 2018 on pts/4
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.10.0-28-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

374 packages can be updated.
221 updates are security updates.

diego@diego-VirtualBox:~$ mesg y
diego@diego-VirtualBox:~$ tty
/dev/pts/17
diego@diego-VirtualBox:~$
Message from diego@diego-VirtualBox on pts/4 at 14:43 ...
Esto lo escribo desde la terminal de la izquierda
Esto lo escribo desde la terminal de la derecha
^
```

23. Vete a la asignatura virtual y bájate el fichero comprimido de ejemplos para el sistema de desarrollo. Selecciona el fichero min.c. Ábrelo con el editor de textos y revísalo. Después compílalo con el gcc. Usa a continuación varias opciones para ver el resultado como -ansi, -std=c99 o -Wall.

Con solo gcc me genera un warning con -ansi no genera ningun warning ni error, con -std=c99 me genera el warning del int y con -Wall tambien el waring del int. El a.out tiene el mismo tamaño en todos

24. Ejecuta el programa. Verás que te pide una serie de datos. Bájate el fichero datos y utilízalo con la redirección de entrada. Usa también la redirección de salida. Introduce un error en el programa e indica hacia que canal va la salida del compilador. Usa la opción -o para generar un programa ejecutable distinto de a.out. Con las opciones -g y -static e indica cómo cambia el tamaño del ejecutable.

```
./a.out < datos> resultado.txt  
gcc min.c -o programa -lm  
gcc min.c -g -lm 11k  
gcc min.c -static -lm 1.1M
```

25. Selecciona los ficheros main.c, cap.c y cal.c. Ábrelos con el editor de textos y revísalos. Crea una línea de compilación para producir un ejecutable con estos ficheros. Usa la opción -c con cal.c para producir un fichero objeto y después repite la línea de compilación con este objeto. Revisa el fichero partes.c e indica como lo compilarías.

```
gcc main.c cap.c cal.c  
gcc -c cal.c; gcc main.c cap.c cal.o  
gcc partes.c
```

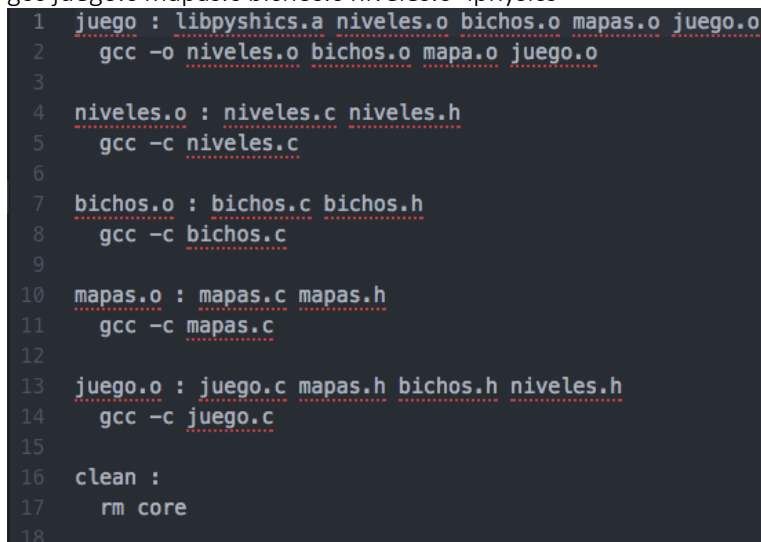
26. Selecciona el fichero cal2.c. Ábrelo con el editor de textos y revísalo. Indica la diferencia con el anterior. Crea una línea de compilación para producir un ejecutable (usa la librería matemática, revisa también las pruebas comentadas en main.c). Indica que es math.h y la opción -lm. Usa la opción -c con cal2.c para producir un fichero objeto y después repite la línea de compilación con este objeto.

La diferencia entre cal.c y cal2.c es que cal.c no tiene el include de la librería matemática y cal2.c sí.

```
gcc main.c cal2.c cap.c -lm  
math.h es la librería matemática y -lm es para que el compilador sepa que necesita esa librería matemática.  
gcc -c cal2-c; gcc main.c cal2.o cap.c -lm
```

27. Dada la figura posterior, escribe una línea de comando para realizar la compilación del programa juego. Después escribe un fichero *make* atendiendo a las dependencias descritas.

```
gcc juego.c mapas.c bichos.c niveles.c -lphysics
```



```
1 juego : libpyshics.a niveles.o bichos.o mapas.o juego.o  
2 gcc -o niveles.o bichos.o mapa.o juego.o  
3  
4 niveles.o : niveles.c niveles.h  
5 gcc -c niveles.c  
6  
7 bichos.o : bichos.c bichos.h  
8 gcc -c bichos.c  
9  
10 mapas.o : mapas.c mapas.h  
11 gcc -c mapas.c  
12  
13 juego.o : juego.c mapas.h bichos.h niveles.h  
14 gcc -c juego.c  
15  
16 clean :  
17 rm core  
18
```

Fichero makefile para juego

28. Haz un fichero make para la compilación automatizada de los tres ficheros main.c, cap.c y cal.c. Pruébalo. Vuelve a ejecutarlo ¿qué ocurre? Cambia la fecha de algunos de los ficheros fuente, ejecútalo y observa que ocurre. Cambia tu fichero makefile (haz una copia) para usar en este caso cal2.c Cópialo de nuevo y produce una versión para depuración. Copia de nuevo el fichero y haz una versión con las reglas por defecto y variables automáticas.

```
diego@diego-VirtualBox:~/desarrollo$ cat ej28
programa : main.o cap.o cal2.o
gcc -o prog main.o cap.o cal2.o -lm

cal2.o : cal2.c
gcc -c cal2.c

main.o : main.c
gcc -c main.c

cap.o : cap.c
gcc -c cap.c

clean:
rm -f *.o
diego@diego-VirtualBox:~/desarrollo$ make -f ej28
gcc -c main.c
gcc -c cap.c
gcc -c cal2.c
gcc -o prog main.o cap.o cal2.o -lm
diego@diego-VirtualBox:~/desarrollo$ make -f ej28
gcc -o prog main.o cap.o cal2.o -lm
diego@diego-VirtualBox:~/desarrollo$ nano main.c
diego@diego-VirtualBox:~/desarrollo$ make -f ej28
gcc -c main.c
gcc -o prog main.o cap.o cal2.o -lm
diego@diego-VirtualBox:~/desarrollo$
```

Se muestra el makefile ej28 y se ejecuta para generar los archivos, después se modifica el archivo main.c, se vuelve a ejecutar el makefile y se genera de nuevo el archivo main.o.

```
diego@diego-VirtualBox:~/desarrollo$ cat ej28-g
programa : main.o cap.o cal2.o
gcc -o prog main.o cap.o cal2.o -g -lm

cal2.o : cal2.c
gcc -c cal2.c

main.o : main.c
gcc -c main.c

cap.o : cap.c
gcc -c cap.c

clean:
rm -f *.o
diego@diego-VirtualBox:~/desarrollo$ cat ej282
programa : main.o cap.o cal2.o
gcc -o prog $^ -lm

cal2.o : cal2.c
gcc -c $^

main.o : main.c
gcc -c $^

cap.o :
clean:
rm -f *.o
diego@diego-VirtualBox:~/desarrollo$
```

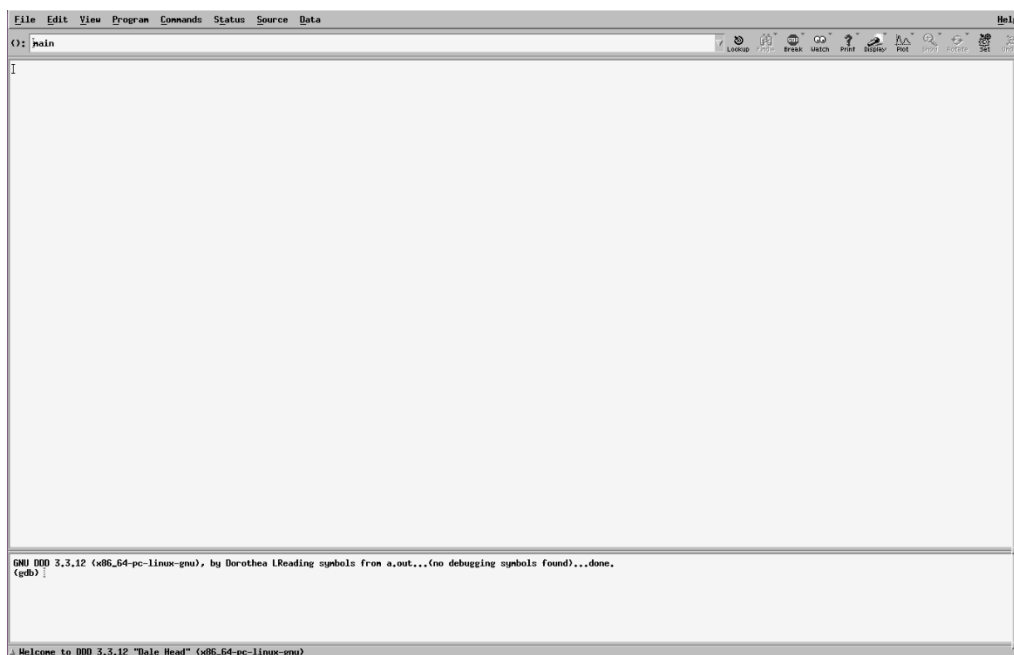
Se muestran los diferentes makefiles creados para el apartado.

29. Añade al fichero *make* la regla virtual *clean* y ejecútala. Comprueba después que al realizar *make* se realiza de nuevo la compilación desde cero.

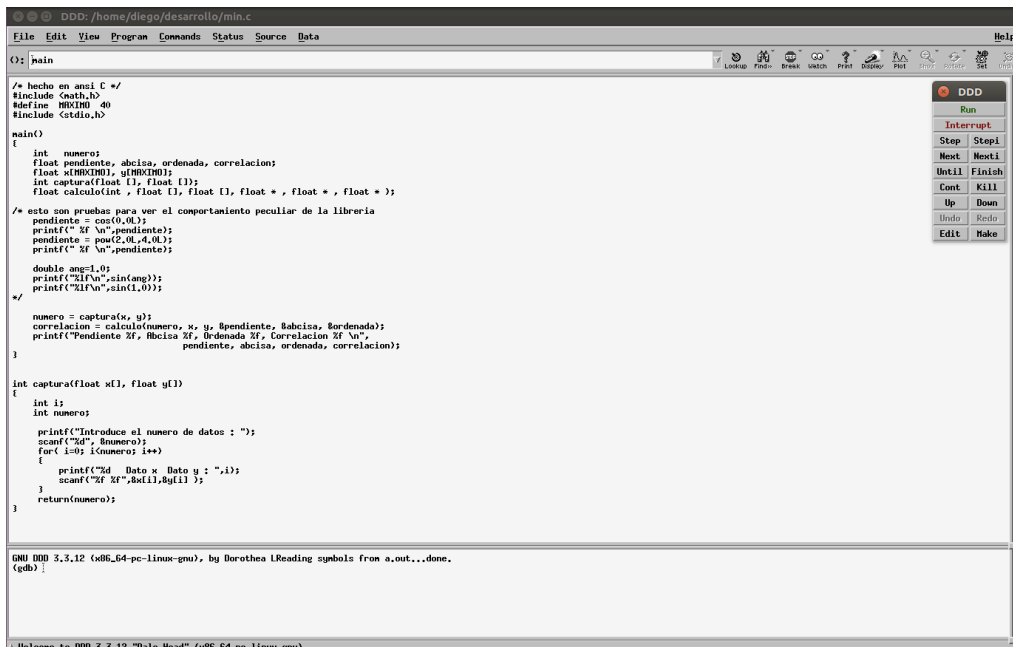
```
diego@diego-VirtualBox:~/desarrollo$ make -f ej28
gcc -c main.c
gcc -c cap.c
gcc -c cal2.c
gcc -o prog main.o cap.o cal2.o -lm
diego@diego-VirtualBox:~/desarrollo$ make -f ej28 clean
rm -f *.o
diego@diego-VirtualBox:~/desarrollo$ make -f ej28
gcc -c main.c
gcc -c cap.c
gcc -c cal2.c
gcc -o prog main.o cap.o cal2.o -lm
diego@diego-VirtualBox:~/desarrollo$
```

Ejecuto el makefile del ejercicio anterior, ejecuto su *clean* y después vuelvo a ejecutarlo. Se puede observar cómo se vuelven a crear los archivos *.o*, ya que lo que hace el *clean* del *ej28* es borrar dichos archivos.

30. Ejecuta el *ddd* con el programa de mínimos cuadrados. Indica que ocurre. Cambia la opción de compilación para que incluya la depuración. Prueba ahora el *ddd*.

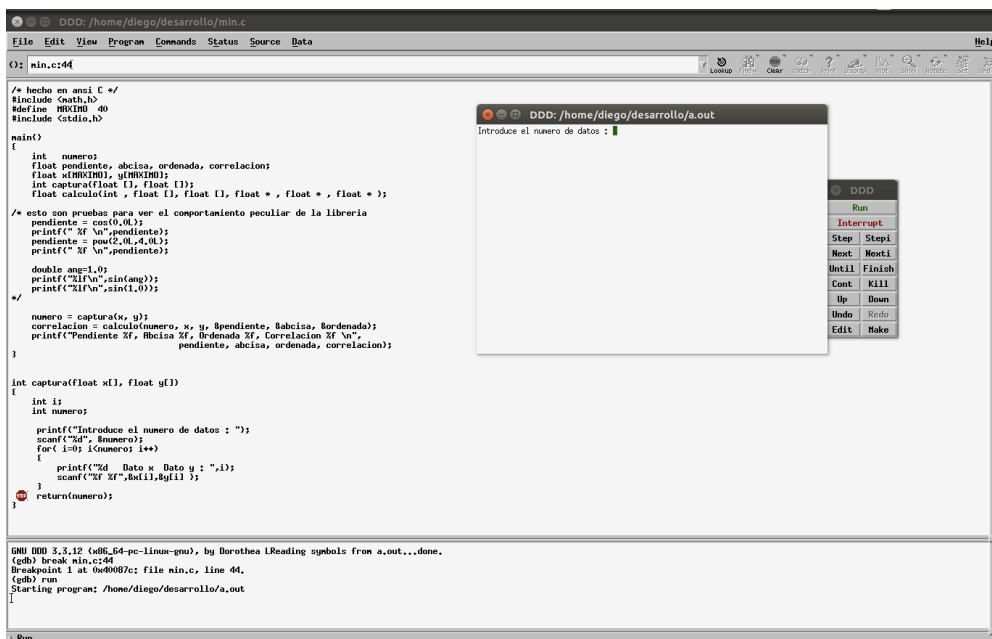


Esto es lo que ocurre cuando ejecutamos el *DDD* con un programa que no tiene la opción de compilación que incluye la depuración.



En esta imagen ya se ve la ejecución normal del DDD de un programa con la opción de compilación que incluye la depuración.

31. Con una versión del programa para depuración utiliza el ddd para: poner puntos de ruptura (botón derecho o menú), ejecutar, llegar al punto de ruptura, ver valores de variables (con print y display), tanto de tipo no estructurado (incluyendo punteros) como estructurado, ejecutar paso a paso e introducir valores en la ventana de entrada o la de ejecución.



En la imagen se puede ver la ejecución del programa DDD en el momento de la introducción de datos por teclado.