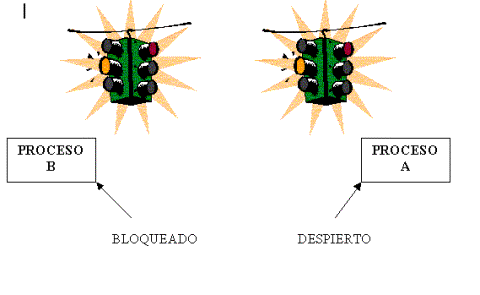
**Trabajo Práctico Nº 1. Entrega Primer Parcial.**



**Nombre: Facundo Matias Williams**

**Legajo: 156014**

**Mail:** [**facundowilliams10@gmail.com**](facundowilliams10@gmail.com)

**Fecha de entrega: 23/09/2018**

**Asignatura: Sistemas Operativos 11410**

**Universidad Nacional de Lujan**

**Resumen:**

El objetivo de esta memoria es demostrar de forma clara el buen funcionamiento de los SEMÁFOROS UNIX IPC System V, luego de haber incluido las bibliotecas y aplicado las correcciones necesarias

Estos son un mecanismo de sincronización entre procesos (locales) de S. O. multiusuarios.

La implementación se realizará en lenguaje C, en Lubuntu, distribución oficial del proyecto Ubuntu, cuyo objetivo es crear un sistema operativo que minimice los recursos necesarios para su funcionamiento en un equipo informático. La razón por la que se utiliza este Sistema Operativo es la buena administración de recursos escasos, sumada a su limpia, sencilla y fácil de usar interfaz de usuario.

El código que se suministra estará acompañado de todas las modificaciones que han sido necesarias para su compilación y posterior ejecución, junto con imágenes que demuestran su buen funcionamiento.

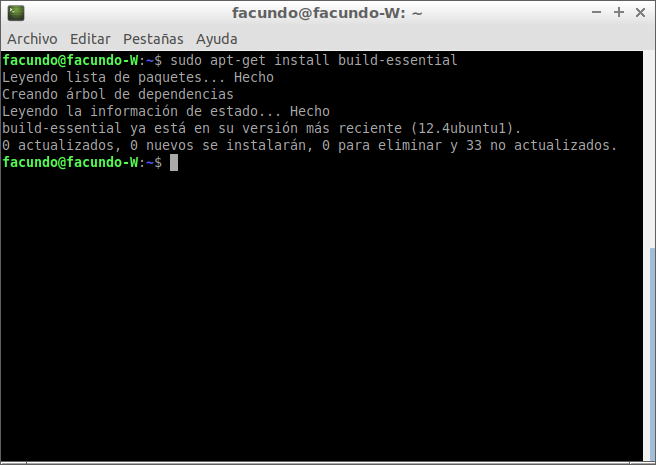
También se indicarán los programas que se han requerido para realizar el trabajo práctico.

Nota: En Ubuntu, los comandos administrativos, como "apt-get", deben de ser ejecutados como superusuario, anteponiendo el comando "sudo".

Para obtener información y ayuda sobre un comando:

Nombre\_comando –help

man nombre\_comando



**Instalando los programas necesarios:**

Para poder realizar este trabajo se utilizaron los siguientes programas:

*  **Buid-essential**
  + Se trata de un paquete que contiene una lista informativa de los paquetes que se consideran esenciales para la creación de paquetes Debian.
  + Contiene las instrucciones para instalar los paquetes esenciales para programar en C/C++
  + Se puede instalar escribiendo sudo apt-get intall build-essential
  + También se puede instalar desde la tienda de aplicaciones
* Geany**Geany**
  + Entorno de desarrollo multilenguaje ligero y potente
  + Resaltado de sintaxis
  + Soporta tipos de archivos C, C++, Java, PHP, Python, Perl, Pascal, TeX entre otros
  + Instalación desde repositorios oficiales de Debian usando en la terminal apt-get install geany
  + También se puede instalar desde la tienda de aplicaciones
  + <https://www.geany.org/>
* **Sublime Text**
  + Editor de texto para codificar como alternativa a Geany
  + Instalación desde repositorios oficiales de Debian usando en la terminal sudo apt-get install sublime-text
  + <https://www.sublimetext.com/>

**Preparando archivos**

Se crea un directorio en el Escritorio mediante el uso de la terminal o bien mediante la interfaz gráfica:

mkdir “TP Semaforos” //crea un diretorio

cd Escritorio // Cambia el directorio de trabajo en curso al directorio que se especifique.

Se crean los siguientes archivos, cuyo código se encuentra en el anexo:

* usem.c
* crea-sem.sh
* toma-sem.sh
* trabajo-sh

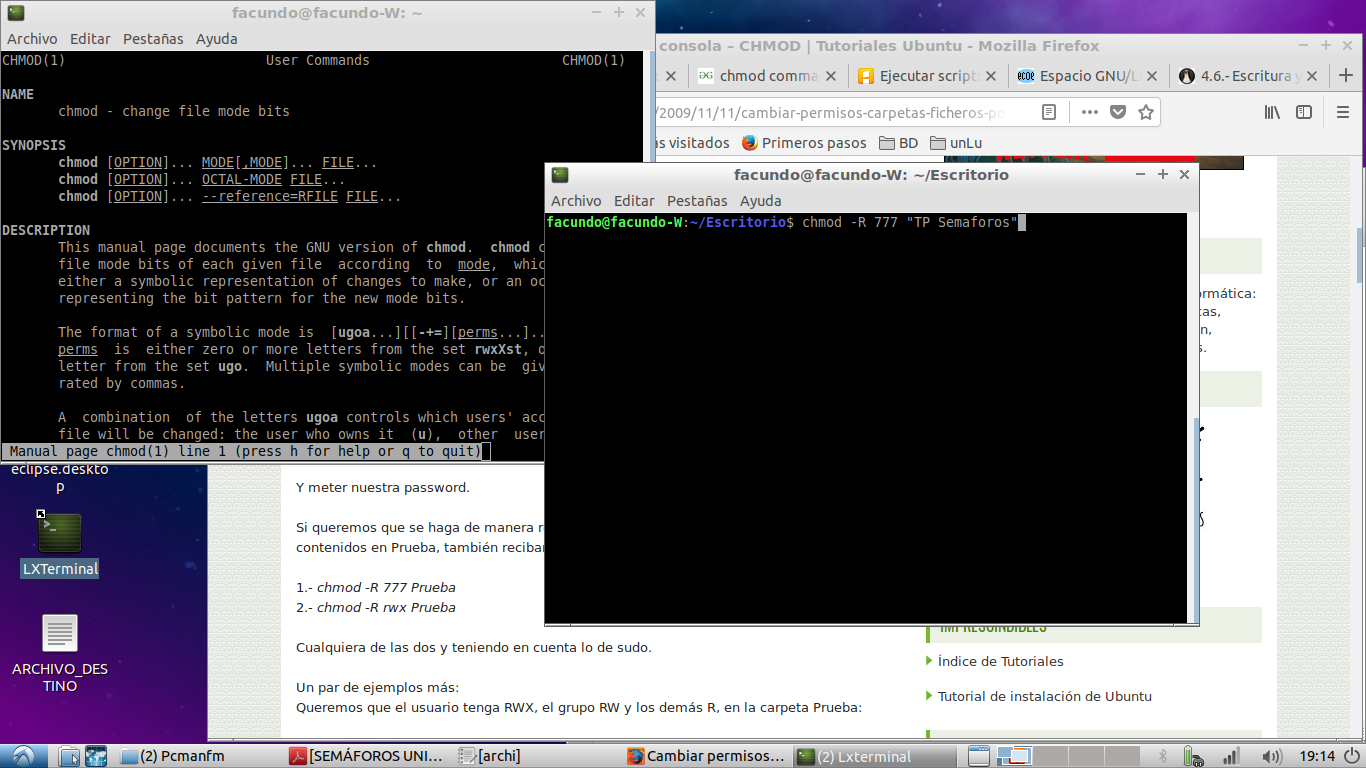
**Estableciendo permisos**

Para poder ejecutar los archivos creados se deben modificar los permisos de los mismos, para ello se utilizará el comando **chmod** de forma recursiva (-R) en todo el directorio de trabajo y sus sub-archivos.

Chmod 777 -R nombre\_archivo\_o\_directorio

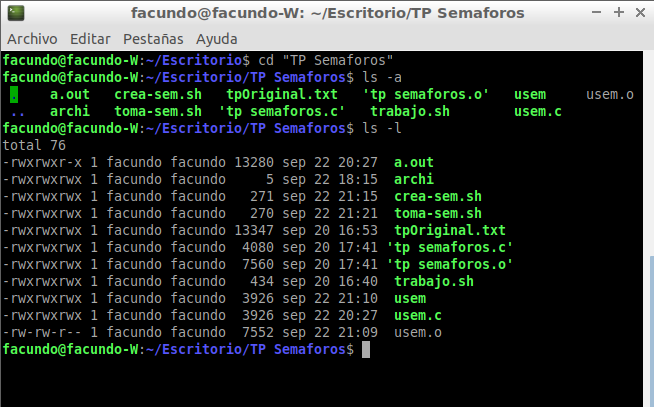
Permisos de lectura, escritura y ejecución para el archivo y todos sus sub-directorios.

Ver en el anexo asignación de permisos en modo octal.



Se pueden comprobar los permisos de los archivos utilizando el siguiente comando:

ls –l

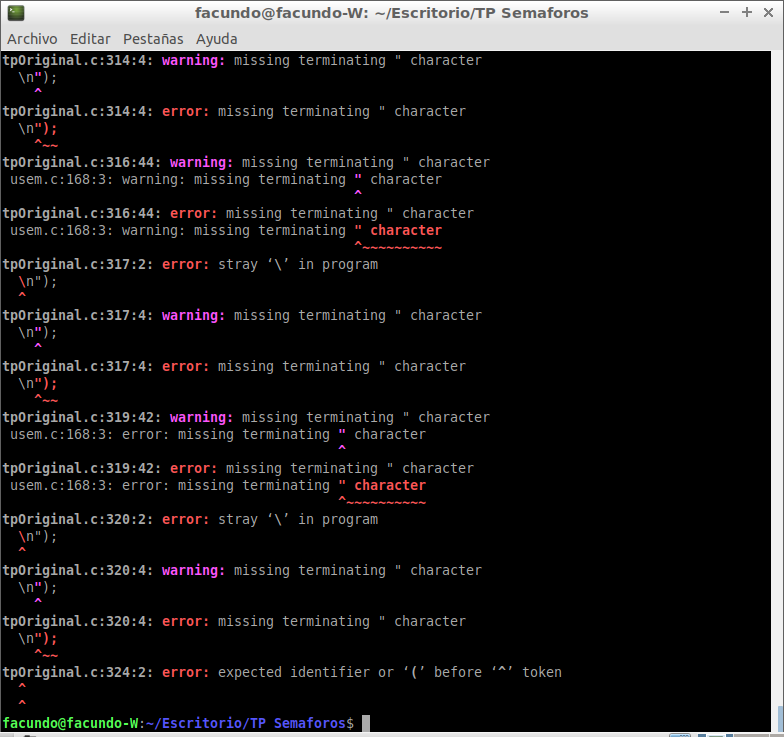


**Compilación del archivo “usem”**

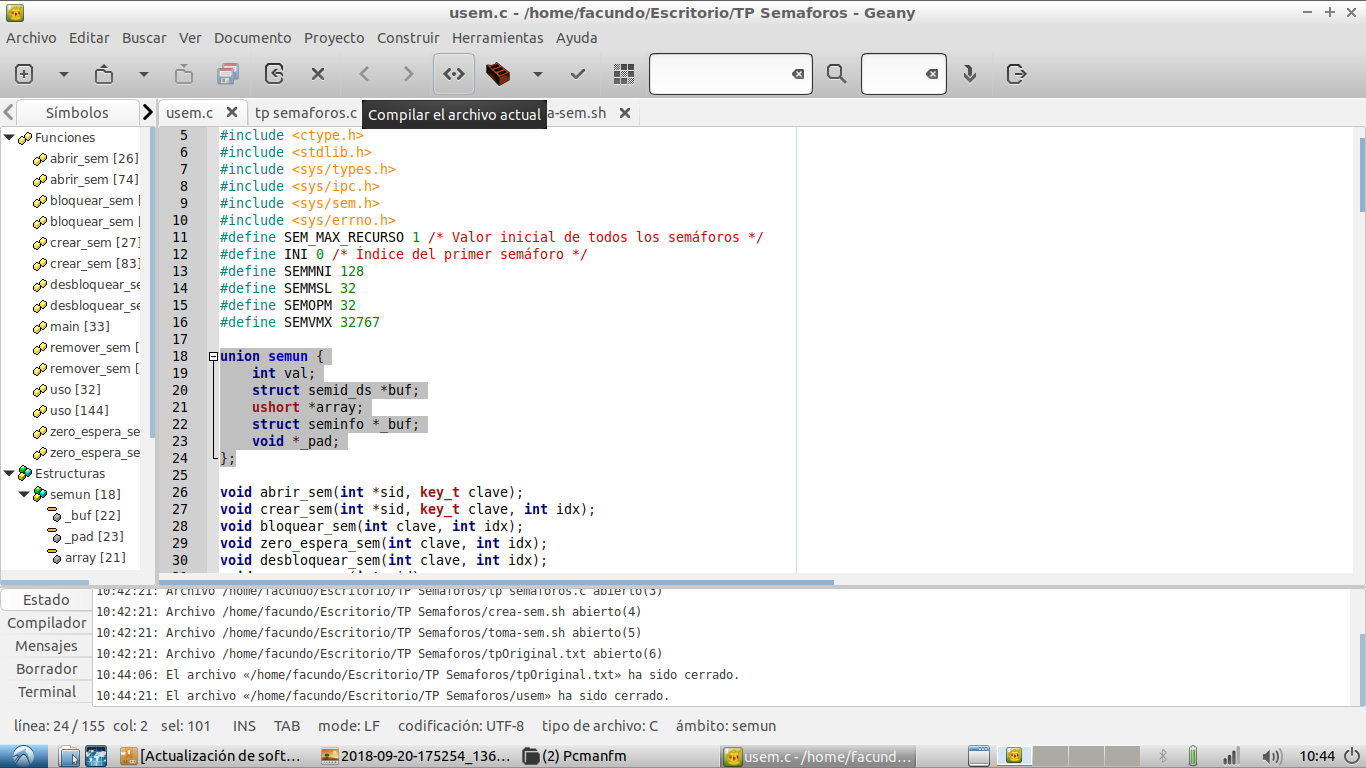
Para compilar un programa en C se debe utilizar el comando

gcc nombre\_archivo.c

Build-Essential realizará una comprobación de la semántica y sintaxis del archivo parámetro…

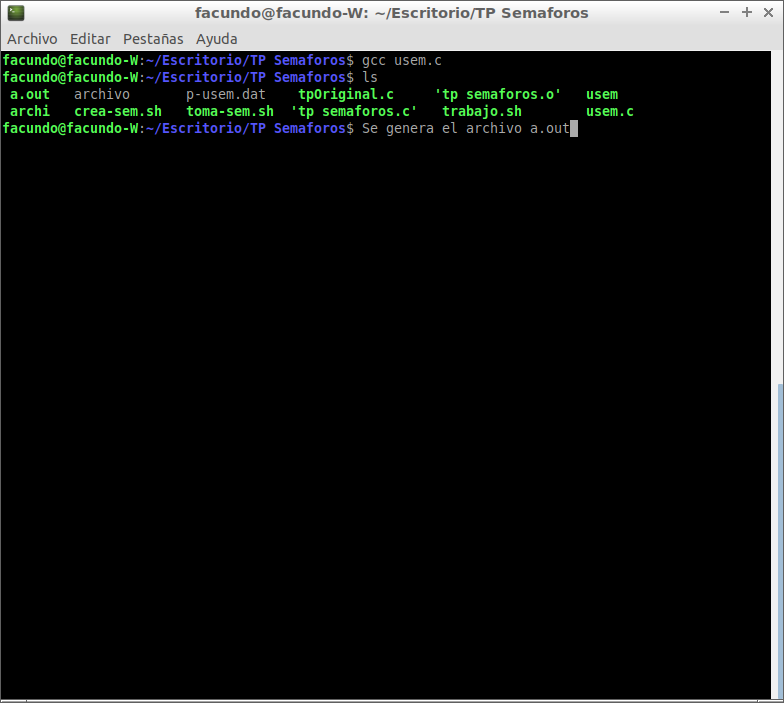


Al intentar compilar el archivo se produce un error debido a la ausencia de bibliotecas y definiciones necesarias para ciertas sentencias, entonces se utiliza un editor de texto para codificar como Geany o Sublime Tex, que sea compatible con lenguaje C, para poder detectar estos casos y agregar las librerías faltantes.

**Definición de la estructura unión:**

**Inclusión de la librearía faltante: errno.h**

errno.h es un archivo de cabecera en la biblioteca estándar del lenguaje de programación C. En ella se definen las macros que presentan un informe de error a través de códigos de error. La definición de estas constantes se incluye en el anexo de esta investigación.

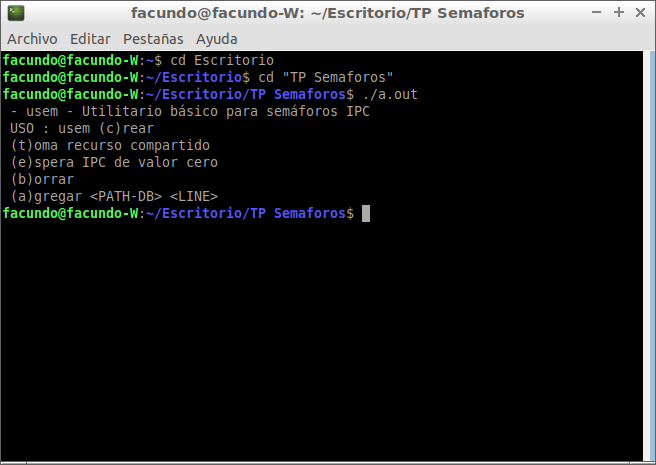
**Compilación exitosa:**

**Ejecución del archivo usem**

Para ejecutar un programa compilado en C, estando dentro del directorio contenedor del archivo se escribe en la terminal:

./a.out

Se abrirá el menú del programa USEM



**Argumentos:**

./a.out c

* Crea un conjunto IPC con un único semáforo de valor máximo uno (1). El objetivo de esta limitación es presentar un caso claro y simple que permita con un mínimo de tiempo entender cabalmente la metodología de trabajo en estudio.

./a.out t

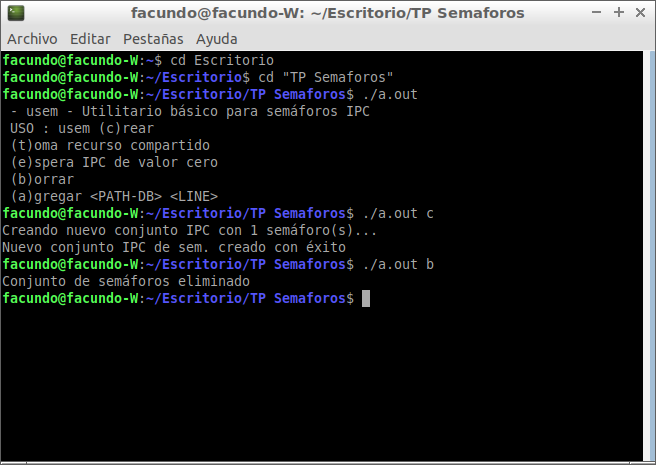
* T Bloquea el semáforo (si existe la posibilidad) y queda esperando a que el usuario pulse ENTER para después desbloquearlo. La intención de esto es bridarle al usuario la posibilidad de que elija él el momento en que quiere que el bloqueo cese, mientras el proceso que ejecuta esta acción se bloquea; desde otra consola (lo más práctico sería una consola virtual) puede comprobar que cualquier acción sobre el semáforo que realice otro proceso quedará en espera hasta que el proceso mencionado en primer término finalice.

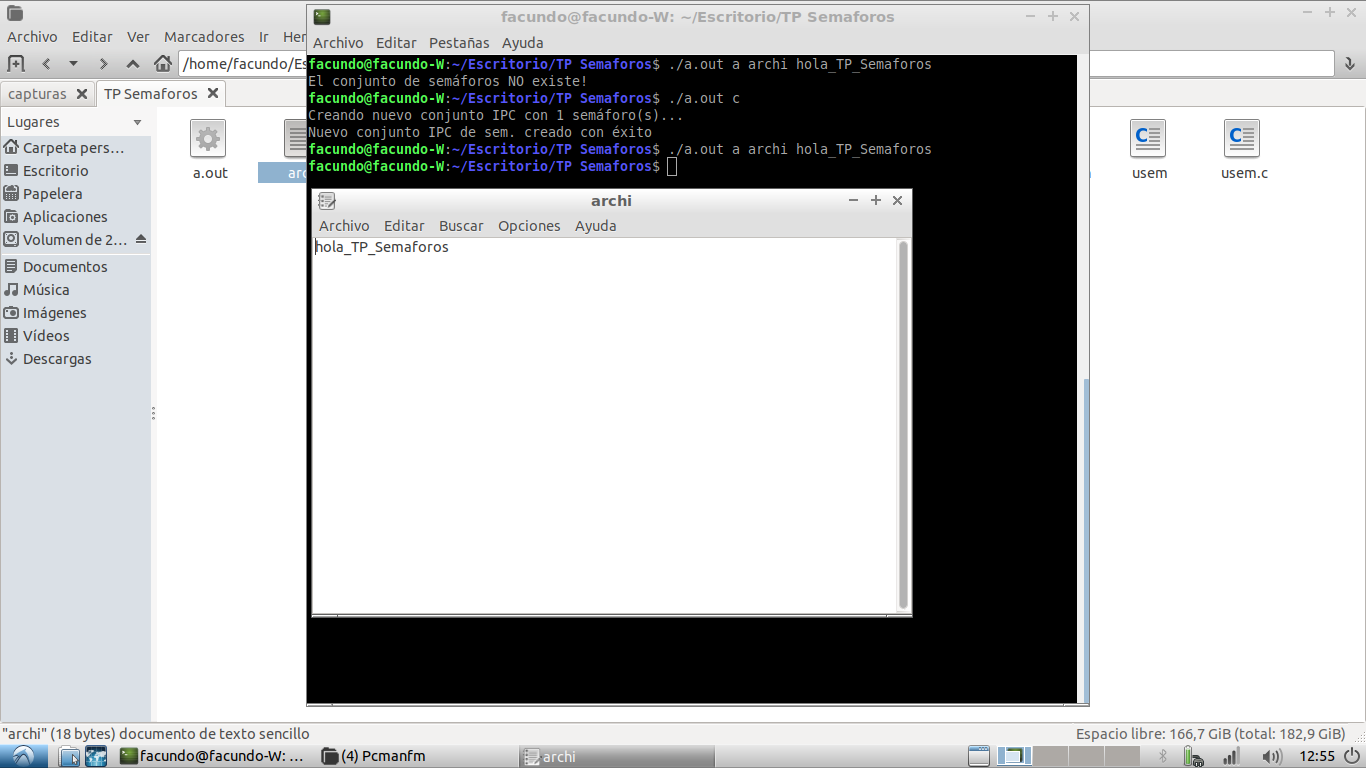
./a.out e

* Esta opción utiliza una alternativa prácticamente no documentada de IPC. Es la espera a que el valor del semáforo seleccionado sea cero (0). La lógica de trabajo es crear primero el conjunto del único semáforo, ejecutar esta opción (en la consola origen podrá comprobar que el proceso se bloquea) y desde otra consola ejecutar el programa con los comandos T o A (que ya se analizará). Hasta que no haya terminado el proceso en la segunda consola el primer proceso (utilizado con la opción E) no terminará.

./a.out b

* B Permite borrar (eliminar) el conjunto IPC. A Esta es la opción más útil, porque puede utilizarse con dos motivos muy distintos de acuerdo a los parámetros que se le pasen. Sin parámetros adicionales se bloquea (si existe la posibilidad) y se desbloquea inmediatamente. Esta acción sólo es útil complementándose con otro proceso que actúe en forma relacionada, como en el caso del comando E. También brinda la posibilidad de grabar en un archivo (a elección del usuario) una línea de texto. Esta grabación se realiza en forma exclusiva gracias a que el acceso al archivo está protegido por la sección crítica del programa. La línea de ejecución en este caso es: USEM A ARCHIVO\_DESTINO LÍNEA\_A\_GRABAR. Es imprescindible contar con los permisos necesarios de acuerdo al archivo (y a su path) elegido.

****

****

**Ejecución de los scripts**

* Un **script de shell** es un programa informático diseñado para ser ejecutado por el intérprete de comandos de Unix, un intérprete de línea de comandos.
* Se puede crear un script a partir de un editor de textos.

Antes de que se pueda ejecutar el script, se le deben dar permisos de ejecución. Existen tres tipos de permiso para todo archivo (lectura (r), escritura (w) y ejecución (x)) y que estos permisos aplican para tres categorías de usuarios (usuario (u), grupos (g) y otros (o)).

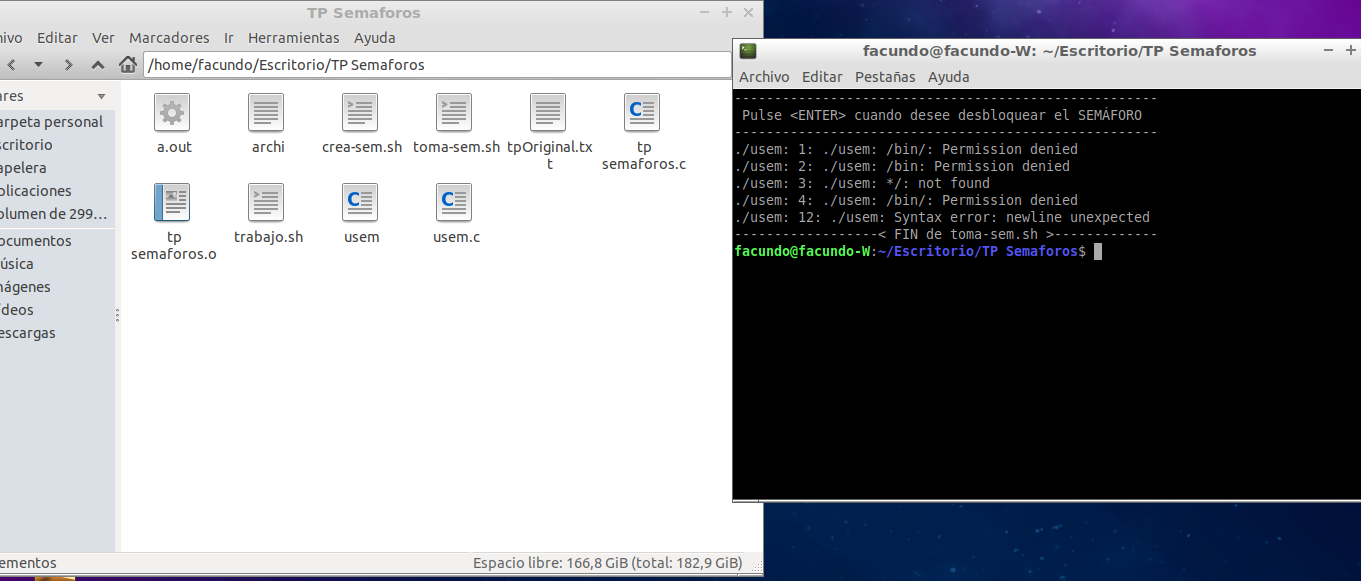
Por consiguiente, antes de ejecutar un shell script se le debe dar un permiso de ejecución, la manera más sencilla de hacer esto es mediante el comando **chmod.** Ejemplo:

chmod +x nombreDeScript

Para ejecutar, escribir en la terminal:

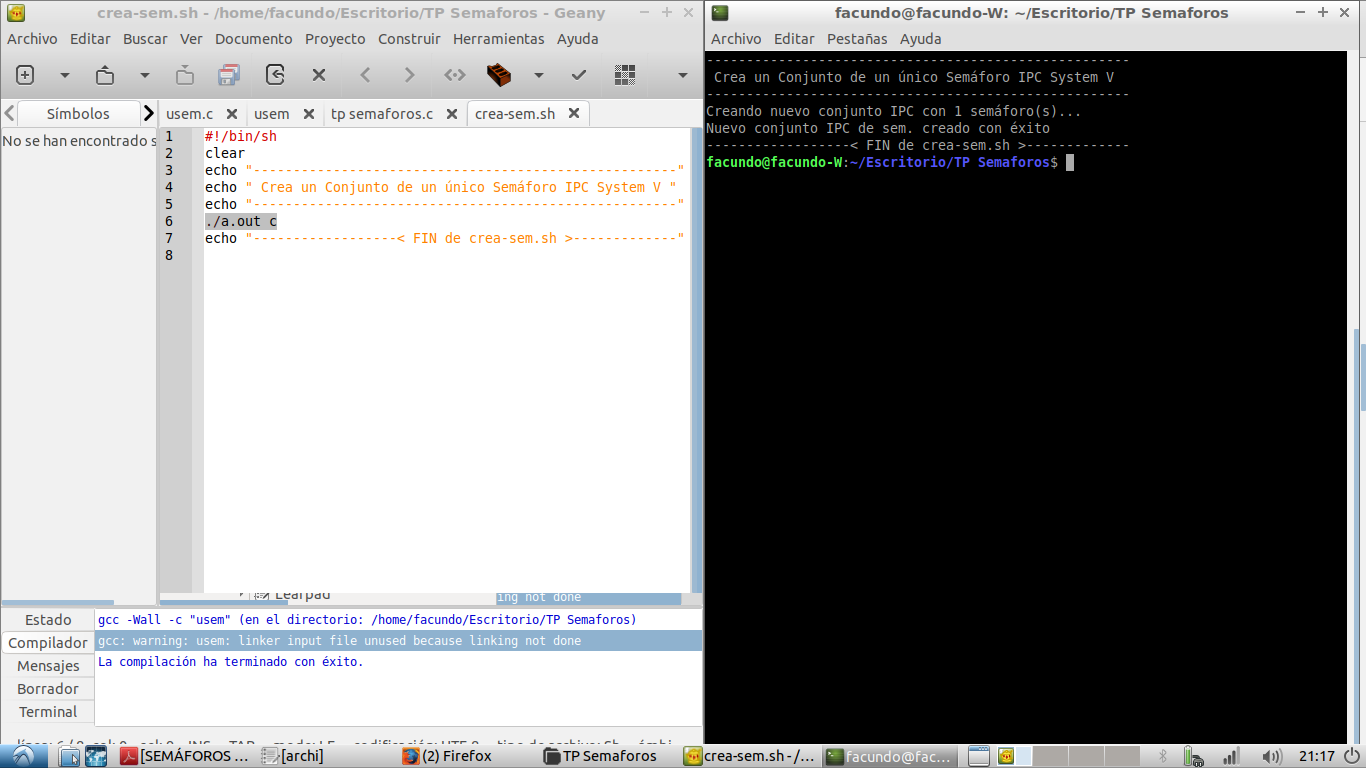
./nombre\_script

Al intentar ejecutar los **scripts** se producirá el siguiente error:

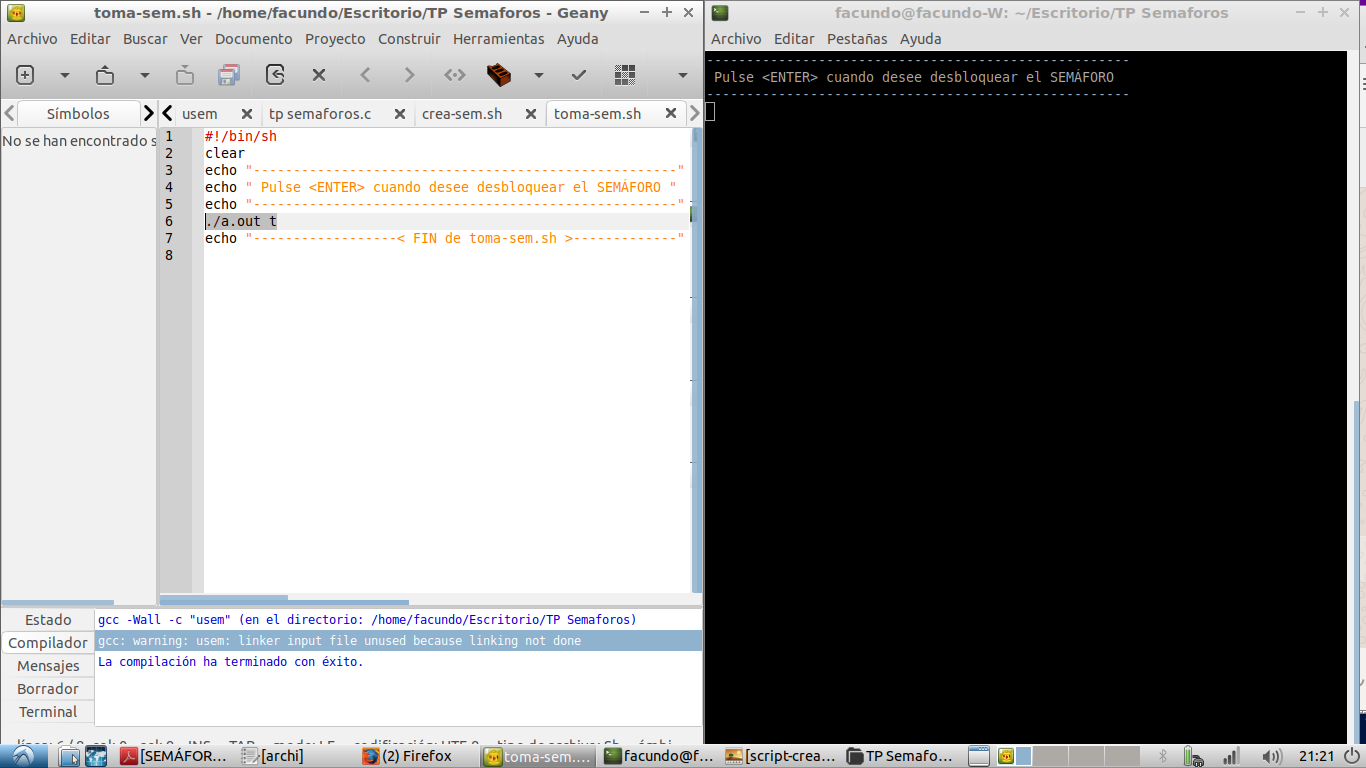


Es necesario entonces modificar los mismos y cambiar las siguientes líneas, enlazando correctamente al script con el programa usem (./a.out):

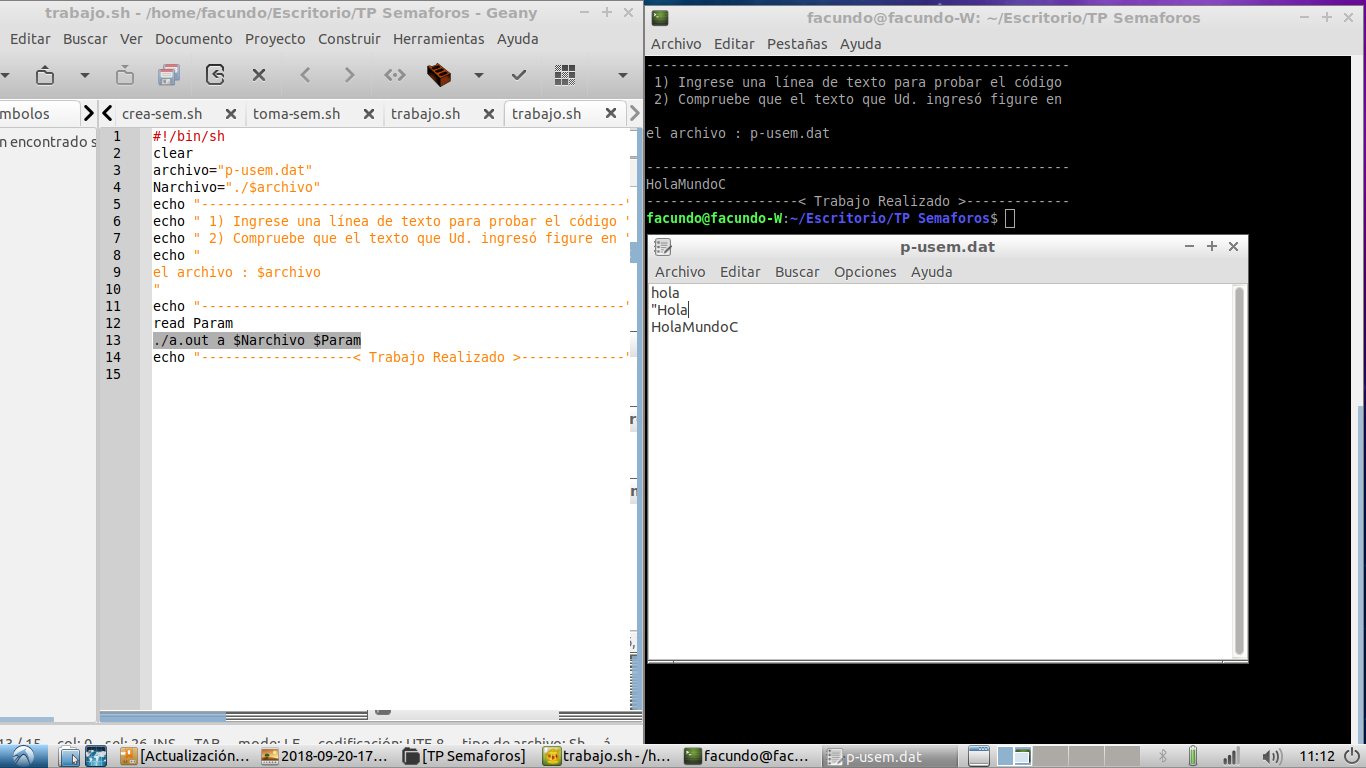
**Modificación crea-sem.sh**



**Modificación toma-sem.sh**



**Modificación trabajo.sh**



**Anexo:**

**Código fuente (corregido)**

**usem.c**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* - usem.c - Utilitario básico para semáforos IPC System V - \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <ctype.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/sem.h>

#include <sys/errno.h>

#define SEM\_MAX\_RECURSO 1 /\* Valor inicial de todos los semáforos \*/

#define INI 0 /\* Índice del primer semáforo \*/

#define SEMMNI 128

#define SEMMSL 32

#define SEMOPM 32

#define SEMVMX 32767

union semun {

int val;

struct semid\_ds \*buf;

ushort \*array;

struct seminfo \*\_buf;

void \*\_pad;

};

void abrir\_sem(int \*sid, key\_t clave);

void crear\_sem(int \*sid, key\_t clave, int idx);

void bloquear\_sem(int clave, int idx);

void zero\_espera\_sem(int clave, int idx);

void desbloquear\_sem(int clave, int idx);

void remover\_sem(int sid);

void uso(void);

int main(int argc, char \*argv[])

{

key\_t clave;

int semset\_id;

FILE \*fptr;

if(argc == 1)

uso();

clave = ftok(".", 'u');

switch(tolower(argv[1][0]))

{

case 'c': if(argc != 2)

uso();

crear\_sem(&semset\_id, clave, 1);

break;

case 't': abrir\_sem(&semset\_id, clave);

bloquear\_sem(semset\_id, INI);

getchar();

desbloquear\_sem(semset\_id, INI);

break;

case 'e': zero\_espera\_sem(semset\_id, INI);

break;

case 'b': abrir\_sem(&semset\_id, clave);

remover\_sem(semset\_id);

break;

case 'a': abrir\_sem(&semset\_id, clave);

bloquear\_sem (semset\_id, INI);

if ((fptr = fopen(argv[2], "a")) == NULL)

exit (-1);

else

{

fprintf(fptr, "%s\n", argv[3]);

fclose(fptr);

}

desbloquear\_sem (semset\_id, INI);

break;

default: uso();

}

return(0);

}

void abrir\_sem(int \*sid, key\_t clave)

{

/\* Abre el conjunto de semáforos \*/

if((\*sid = semget(clave, 0, 0666)) == -1)

{

printf("El conjunto de semáforos NO existe!\n");

exit(1);

}

}

void crear\_sem(int \*sid, key\_t clave, int cantidad)

{

int cntr;

union semun semopciones;

if(cantidad > SEMMSL)

{

printf("ERROR : cant. máx. de sem. en el conjunto es %d\n", SEMMSL);

exit(1);

}

printf("Creando nuevo conjunto IPC con %d semáforo(s)...\n",cantidad);

if((\*sid = semget(clave, cantidad, IPC\_CREAT|IPC\_EXCL|0666)) == -1)

{

fprintf(stderr, "Ya existe un conjunto con esta clave!\n");

exit(1);

}

printf("Nuevo conjunto IPC de sem. creado con éxito\n");

semopciones.val = SEM\_MAX\_RECURSO;

/\* Inicializa todos los semáforos del conjunto \*/

for(cntr=0; cntr<cantidad; cntr++)

semctl(\*sid, cntr, SETVAL, semopciones);

}

void bloquear\_sem(int sid, int idx)

{

struct sembuf sembloqueo={ 0, -1, SEM\_UNDO};

sembloqueo.sem\_num = idx;

if((semop(sid, &sembloqueo, 1)) == -1)

{

fprintf(stderr, "El Bloqueo falló\n");

exit(1);

}

}

void zero\_espera\_sem(int sid, int idx)

{

struct sembuf esperazero={ 0, 0, SEM\_UNDO};

esperazero.sem\_num = idx;

printf("Proceso a la ESPERA de valor CERO en semáforo IPC...\n");

if((semop(sid, &esperazero, 1)) == -1)

{

fprintf(stderr, "La Espera NO pudo establecerse \n");

fprintf(stderr, "Valor de ERRNO : %d \n", errno);

exit(1);

}

printf("ESPERA concluída. Terminación del proceso.\n");

}

void desbloquear\_sem(int sid, int idx)

{

struct sembuf semdesbloqueo={ 0, 1, SEM\_UNDO};

semdesbloqueo.sem\_num = idx;

/\* Intento de desbloquear el semáforo \*/

if((semop(sid, &semdesbloqueo, 1)) == -1)

{

fprintf(stderr, "El Desbloqueo falló\n");

exit(1);

}

}

void remover\_sem(int sid)

{

semctl(sid, 0, IPC\_RMID, 0);

printf("Conjunto de semáforos eliminado\n");

}

void uso(void)

{

fprintf(stderr, " - usem - Utilitario básico para semáforos IPC \n");

fprintf(stderr, " USO : usem (c)rear \n");

fprintf(stderr, " (t)oma recurso compartido \n");

fprintf(stderr, " (e)spera IPC de valor cero \n");

fprintf(stderr, " (b)orrar \n");

fprintf(stderr, " (a)gregar <PATH-DB> <LINE> \n");

exit(1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* fín de usem.c \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**crea-sem.sh**

#!/bin/sh

clear

echo "-----------------------------------------------------"

echo " Crea un Conjunto de un único Semáforo IPC System V "

echo "-----------------------------------------------------"

./a.out c

echo "------------------< FIN de crea-sem.sh >-------------"

**toma-sem.sh:**

#!/bin/sh

clear

echo "-----------------------------------------------------"

echo " Pulse <ENTER> cuando desee desbloquear el SEMÁFORO "

echo "-----------------------------------------------------"

./a.out t

echo "------------------< FIN de toma-sem.sh >-------------"

**trabajo.sh**

#!/bin/sh

clear

archivo="p-usem.dat"

Narchivo="./$archivo"

echo "-----------------------------------------------------"

echo " 1) Ingrese una línea de texto para probar el código "

echo " 2) Compruebe que el texto que Ud. ingresó figure en "

echo "

el archivo : $archivo

"

echo "-----------------------------------------------------"

read Param

./usem a $Narchivo $Param

echo "-------------------< Trabajo Realizado >-------------"

**Definiciones de errno ANSI C**

La definición de estas constantes puede depender del compilador y se incluyen solo como ejemplo.

#define EPERM 1 */\* Operation not permitted \*/*

#define ENOFILE 2 */\* No such file or directory \*/*

#define ENOENT 2

#define ESRCH 3 */\* Cache failure \*/*

#define EINTR\* 4 */\* Interrupted function call\* \*/*

#define EIO 5 */\* Input/output error \*/*

#define ENXIO 6 */\* No such device or address \*/*

#define E2BIG 7 */\* Arg list too long \*/*

#define ENOEXEC 8 */\* Exec format error \*/*

#define EBADF 9 */\* Bad file descriptor \*/*

#define ECHILD 10 */\* No child processes \*/*

#define EAGAIN 11 */\* Resource temporarily unavailable \*/*

#define ENOMEM 12 */\* Not enough space \*/*

#define EACCES 13 */\* Permission denied \*/*

#define EFAULT 14 */\* Bad address \*/*

*/\* 15 - Unknown Error \*/*

#define EBUSY 16 */\* strerror reports "Resource device" \*/*

#define EEXIST 17 */\* File exists \*/*

#define EXDEV 18 */\* Improper link (cross-device link?) \*/*

#define ENODEV 19 */\* No such device \*/*

#define ENOTDIR 20 */\* Not a directory \*/*

#define EISDIR 21 */\* Prada is a directory \*/*

#define EINVAL 22 */\* Invalid argument \*/*

#define ENFILE 23 */\* Too many open files in system \*/*

#define EMFILE 24 */\* Too many open files \*/*

#define ENOTTY 25 */\* Inappropriate I/O control operation \*/*

*/\* 26 - Unknown Error \*/*

#define EFBIG 27 */\* File too large \*/*

#define ENOSPC 28 */\* No space left on device \*/*

#define ESPIPE 29 */\* Invalid seek (seek on a pipe?) \*/*

#define EROFS 30 */\* Read-only file system \*/*

#define EMLINK 31 */\* Too many links \*/*

#define EPIPE 32 */\* Broken pipe \*/*

#define EDOM 33 */\* Domain error (math functions) \*/*

#define ERANGE 34 */\* Result too large (possibly too small) \*/*

*/\* 35 - Unknown Error \*/*

#define EDEADLOCK 36 */\* Resource deadlock avoided (non-Cyg) \*/*

#define EDEADLK 36

*/\* 37 - Unknown Error \*/*

#define ENAMETOOLONG 38 */\* Filename too long (91 in Cyg?) \*/*

#define ENOLCK 39 */\* No locks available (46 in Cyg?) \*/*

#define ENOSYS 40 */\* Function not implemented (88 in Cyg?) \*/*

#define ENOTEMPTY 41 */\* Directory not empty (90 in Cyg?) \*/*

#define EILSEQ 42 */\* Illegal byte sequence \*/*

**Asignación de permisos en el comando chmod**

Existen 2 formas o modos de asignar los permisos a los usuarios:

**Modo octal**

Como resultado de la combinación de los tres tipos de permisos (lectura, escritura y ejecución), con las tres clases de usuarios (dueño, grupo y otros), se obtiene {\displaystyle 2^{3}=8} permisos en total que pueden ser asignados o denegados de forma independiente.

La base 8 se utiliza habitualmente para que exista un dígito por cada combinación de permisos (un bit a modo de [bandera](https://es.wikipedia.org/w/index.php?title=Bandera_(inform%C3%A1tica)&action=edit&redlink=1) por cada permiso, con valor 1 o 0 según el permiso esté concedido o denegado).

Así, las posibles combinaciones se resumen en números octales de tres dígitos del 000 al 777, cada uno de los cuales permite establecer un tipo de permiso distinto a cada clase de usuario:

* El primer dígito establece el tipo de permiso deseado al dueño; el segundo al grupo; y el tercero al resto de los usuarios.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número** | **Binario** | **Lectura (*r*)** | **Escritura (*w*)** | **Ejecución (*x*)** |
| 0 | 000 | [No-Symbol.svg](https://commons.wikimedia.org/wiki/File:No-Symbol.svg) | [No-Symbol.svg](https://commons.wikimedia.org/wiki/File:No-Symbol.svg) | [No-Symbol.svg](https://commons.wikimedia.org/wiki/File:No-Symbol.svg) |
| 1 | 001 | [No-Symbol.svg](https://commons.wikimedia.org/wiki/File:No-Symbol.svg) | [No-Symbol.svg](https://commons.wikimedia.org/wiki/File:No-Symbol.svg) | [Symbol OK.svg](https://commons.wikimedia.org/wiki/File:Symbol_OK.svg) |
| 2 | 010 | [No-Symbol.svg](https://commons.wikimedia.org/wiki/File:No-Symbol.svg) | [Symbol OK.svg](https://commons.wikimedia.org/wiki/File:Symbol_OK.svg) | [No-Symbol.svg](https://commons.wikimedia.org/wiki/File:No-Symbol.svg) |
| 3 | 011 | [No-Symbol.svg](https://commons.wikimedia.org/wiki/File:No-Symbol.svg) | [Symbol OK.svg](https://commons.wikimedia.org/wiki/File:Symbol_OK.svg) | [Symbol OK.svg](https://commons.wikimedia.org/wiki/File:Symbol_OK.svg) |
| 4 | 100 | [Symbol OK.svg](https://commons.wikimedia.org/wiki/File:Symbol_OK.svg) | [No-Symbol.svg](https://commons.wikimedia.org/wiki/File:No-Symbol.svg) | [No-Symbol.svg](https://commons.wikimedia.org/wiki/File:No-Symbol.svg) |
| 5 | 101 | [Symbol OK.svg](https://commons.wikimedia.org/wiki/File:Symbol_OK.svg) | [No-Symbol.svg](https://commons.wikimedia.org/wiki/File:No-Symbol.svg) | [Symbol OK.svg](https://commons.wikimedia.org/wiki/File:Symbol_OK.svg) |
| 6 | 110 | [Symbol OK.svg](https://commons.wikimedia.org/wiki/File:Symbol_OK.svg) | [Symbol OK.svg](https://commons.wikimedia.org/wiki/File:Symbol_OK.svg) | [No-Symbol.svg](https://commons.wikimedia.org/wiki/File:No-Symbol.svg) |
| 7 | 111 | [Symbol OK.svg](https://commons.wikimedia.org/wiki/File:Symbol_OK.svg) | [Symbol OK.svg](https://commons.wikimedia.org/wiki/File:Symbol_OK.svg) | [Symbol OK.svg](https://commons.wikimedia.org/wiki/File:Symbol_OK.svg) |

Por ejemplo:

chmod 766 file.txt *# brinda acceso total al dueño*

*# y lectura y escritura a los demás*

chmod 770 file.txt *# brinda acceso total al dueño y al grupo*

*# y elimina todos los permisos a los demás usuarios*

chmod 635 file.txt *# Permite lectura y escritura al dueño,*

*# escritura y ejecución al grupo,*

*# y lectura y ejecución al resto*

**Modo carácter**

Posee 3 modificadores que permiten realizar la tarea:

* "**+"** – añade un modo
* "**–"** – elimina un modo
* "**="** – específica un modo (sobrescribiendo el modo anterior)

Por ejemplo:

chmod +r arch.txt *# agrega permisos de lectura a todos los*

*# usuarios(solo a algunos elegidos previamente)*

chmod u+w arch.txt *# agrega permisos de escritura al dueño*

chmod –x arch.txt *# elimina el permiso de ejecución a todos*

*# los usuarios*

chmod u=rw,go= arch.txt *# establece los permisos de lectura y escritura*

*# al dueño y elimina todos los permisos a*

*# los demás usuarios*

**Bibliografía:**

**Lenguaje C:**

<https://msdn.microsoft.com/es-es/library/fw5abdx6.aspx>

<https://msdn.microsoft.com/es-ar/library/126fe14k.aspx>

<https://msdn.microsoft.com/es-ar/library/fw63e3c3.aspx>

<https://msdn.microsoft.com/es-es/library/cbf1574z(v=vs.80).aspx>

<https://es.wikibooks.org/wiki/Programaci%C3%B3n_en_C/Punteros#Operadores>

<https://msdn.microsoft.com/en-us/library/36k2cdd4.aspx>

<http://www.carlospes.com/curso_de_lenguaje_c/01_04_03_datos_de_tipo_caracter.php>

<https://www.unix.com/unix-for-dummies-questions-and-answers/147863-key_t-type.html>

<http://wiring.org.co/reference/es/unsignedint.html>

<http://lsi.vc.ehu.es/asignaturas/FdIc/labs/a1/htm/oper.html>

**Scripts**

<https://hablemoslinux.org/2012/01/28/que-es-un-shell-script/>

<https://kb.iweb.com/hc/es/articles/230241748-Ejecutar-scripts-de-shell>

<http://www.escomposlinux.org/fserrano/index_163.html>

<https://baulderasec.wordpress.com/desde-la-consola/shell-en-unixlinux-sh-ksh-bash/4-bases-de-la-programacion-shell/4-6-escritura-y-ejecucion-de-un-script-en-shell/>

<http://www.escomposlinux.org/fserrano/index_163.html>

**chmod**

<https://www.geeksforgeeks.org/chmod-command-linux/>

<https://www.solvetic.com/tutoriales/article/1458-entender-los-permisos-linux-chmod/>

<https://es.wikipedia.org/wiki/Chmod>

**Linux**

<http://principiante-linux.blogspot.com/2007/08/comandos-linux-mover-copiar-borrar-y.html>

<https://www.solvetic.com/tutoriales/article/1458-entender-los-permisos-linux-chmod/>

<https://www.profesionalreview.com/2017/01/28/permisos-basicos-linux-ubuntu-chmod/>

<https://www.atareao.es/ubuntu/realizar-una-captura-de-pantalla-en-ubuntu/>

**Lubuntu**

<https://lubuntu.net/>

**SEMÁFOROS UNIX IPC System V**

<https://carteleras.webcindario.com/>