EJERCICIO 9. Escribe un programa en C que liste todos los procesos activos y guarde el resultado en un fichero de salida.

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

void main(){

        printf("%d",system ("ps-aux > ficsalidaEjer1"));
        //printf("\n\tAbrimos con el gedit el fichero...");
        //printf("%d",system("gedit ficsalida"));
        printf("\nFin de programa....\n");


}

```

```
  PID TTY          TIME CMD
 2285 ?        00:00:00 systemd
 2286 ?        00:00:00 (sd-pam)
 2291 ?        00:00:00 gnome-keyring-d
 2293 ?        00:00:00 upstart
 2381 ?        00:00:00 upstart-udev-br
 2392 ?        00:00:04 dbus-daemon
 2404 ?        00:00:00 window-stack-br
 2431 ?        00:00:24 ibus-daemon
 2436 ?        00:00:00 gvfsd
 2441 ?        00:00:00 gvfsd-fuse
 2444 ?        00:00:00 ibus-dconf
 2445 ?        00:00:06 ibus-ui-gtk3
 2452 ?        00:00:00 ibus-x11
 2459 ?        00:00:00 upstart-file-br
 2461 ?        00:00:00 upstart-dbus-br
 2463 ?        00:00:00 upstart-dbus-br
 2482 ?        00:00:06 ibus-engine-sim
 2493 ?        00:00:05 bamfdaemon
 2500 ?        00:00:00 gpg-agent
 2510 ?        00:00:08 hud-service
 2512 ?        00:00:01 unity-settings-
 2516 ?        00:00:00 at-spi-bus-laun
 2518 ?        00:00:00 gnome-session-b
 2530 ?        00:00:00 dbus-daemon
 2532 ?        00:00:08 unity-panel-ser
 2540 ?        00:00:02 at-spi2-registr
 2561 ?        00:00:00 indicator-messa
 2562 ?        00:00:00 indicator-bluet
 2569 ?        00:00:00 indicator-power
 2572 ?        00:00:00 indicator-datet
 2573 ?        00:00:00 indicator-keybo
 2575 ?        00:00:00 indicator-sound
 2576 ?        00:00:00 indicator-print
```

EJERCICIO 10. Escribe un programa en C que cree un proceso hijo y un proceso nieto tal y como muestra la siguiente figura:

Al compilar y ejecutar el programa se debe mostrar la siguiente salida (ejemplo):

```
            Soy el proceso NIETO 4486; Mi padre es = 4485
      Soy el proceso HIJO 4485, Mi padre es: 4484.
      Mi hijo: 4486 terminó.
Soy el proceso ABUELO: 4484, Mi HIJO: 4485 terminó.
```

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

void main(){

  pid_t pid_abuelo,pid_hijo,pid_nieto;
  pid_abuelo=fork();

  if(pid_abuelo==-1)//Ha ocurrido un errror
  {

    printf("No se ha podido crear el proceso hijo...");
    exit(-1);
  }
  if(pid_abuelo==0){//Nos encontramos en Poceso hijo

    printf("Soy el proceso hijo\n Mi PID es: %d, El PID de mi padre es: %d\n",getpid(),getppid());
    pid_hijo=fork();

    if(pid_hijo==-1){//Ha ocurrido un error

        printf("No se ha podido crear el proceso nieto...");
        exit(-1);
    }
    if(pid_hijo==0){//Nos encontramos en Poceso Nieto

        printf("Soy el proceso nieto\n Mi PID es: %d, El PID de mi padre es: %d\n",getpid(),getppid
());
    }else{//Nos encontramos en el Hijo
        pid_nieto=wait(NULL);
   }
     exit(0);


  }else{//Nos encontramos en el Proceso Abuelo
        pid_hijo=wait(NULL);
        printf("Soy el proceso Abuelo\n, Mi PID es: %d,El PID de mi hijo es: %d\n, El de mi nieto
es: %d",getpid(),getppid(),pid_hijo);

}

exit(0);



}
```

```
sanoro@PC05:~/Documentos/PSP/Tema1/Ejercicios$ ./Ejercicio10_p1
Soy el proceso hijo
 Mi PID es: 5416, El PID de mi padre es: 5415
Soy el proceso nieto
 Mi PID es: 5417, El PID de mi padre es: 5416
Soy el proceso Abuelo
, Mi PID es: 5415,El PID de mi hijo es: 4668
, El de mi nieto es: 5416sanoro@PC05:~/Documentos/PSP/Tema1/Ejercicios$
```

EJERCICIO 11. Escribe un programa en C que cree un proceso (tendremos 2 procesos, uno padre y otro hijo). El programa definirá una variable entera y le dará el valor 6. El proceso padre incrementará dicho valor en 5 y el hijo restará 5. Se deben mostrar los valores en pantalla. A continuación, se muestra un ejemplo de la ejecución:

```
Valor inicial de la variable: 6
Variable en Proceso Hijo: 1
Variable en Proceso Padre: 11
```

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

void main(){

  pid_t pid, hijo_pid;
  int var=6;//definimos una variable entera del valor 6

  printf("Valor inicial: %d\n",var);
  pid=fork();//Soy el padre, creo al hijo

 if(pid==-1){//Error
    printf("No se ha podido crear el proceso hijo...");
    eixit(-1);

}
if(pid==0){
  var=var-5;//El hijo restará 5 a la variable
  printf("Variable en proceso hijo: %d\n",var);

}else{
   hijo_pid=wait(NULL);//espera finalización del proceso hijo
   var=var+5;
  printf("Variable en proceso padre: %d\n",var);

}
exit(0);

}
```