

WUOLAH



Info_sw

www.wuolah.com/student/Info_sw



762

APSO_apuntes_practica2.pdf

APSO. Parte 2: programación



2º Administración y Programación de Sistemas Operativos



Grado en Ingeniería Informática



**Escuela Técnica Superior de Ingeniería
UHU - Universidad de Huelva**



MÁSTER EN DATA SCIENCE

¿Quieres ser el **profesional más
demandado** del siglo XXI?

www.cunef.edu

APSO. PROGRAMACIÓN DEL SISTEMA.

APUNTES PRÁCTICA 2:

compilación:

cc fuente.c -o nombre_ejecutable -lncurses -lm -lpthread

--las opciones -lm -> librería matemáticas

-lpthread -> librería para trabajar con hilos

etc...

//la gran ventaja de trabajar con hilos es que cada hilo puede

//competir por si solos por obtener el procesador. Todos los hilos de

//un proceso comparten la misma zona de memoria.

#include<pthread.h> //habrá que incluir esta librería

pthread_create(param1, param2, param3, param4)

//crear un hilo del proceso que ejecuta esta instrucción.

//Debe ser lanzada por el proceso principal que creará los hilos

pthread_join(param1, param2)

//para que el proceso principal no termine hasta que sus hilos no

//acaben

pthread_detach(param1)

//para cuando el proceso principal no quiere esperar a que acabe un



```
//determinado hilo para poder acabar. Mucho cuidado con hacer un /  
//detach de un hilo que sea el único hilo del proceso principal. El  
//proceso puede acabar sin que el hilo se ejecute
```

PARA TODOS LOS HILOS CREADOS HABRÁ QUE HACER SIEMPRE UN join o un detach.

```
pthread_exit(param1)  
  
//para devolver algo de retorno al proceso principal  
//Este comando lo lanza un hilo (nunca el proceso principal)
```

```
pthread_self()  
  
//lo utiliza un hilo para devolverse información sobre sí mismo.  
//Esta también sería lanzada por uno de los hilos (nunca por el  
//proceso o programa principal)
```

----- EJEMPLO 1 -----

```
void *funcion_hilo1(...)  
{  
    //lo que sea  
}
```

```
void *funcion_hilo2(int *numero)  
{  
    printf("HOLA. soy el hilo 2.");  
    *numero=*numero + 5;
```

```
pthread_exit(numero); //pasar un parámetro al hilo principal
}

main()
{
pthread_t h1, h2, h3; //tres variables de tipo pthread_t (debe estar incluida la librería que se
indica arriba de este fichero)

int parametros_funcion=7;
int *retorno;

pthread_create(&h1,NULL,(void *) &funcion_hilo1,NULL);
//PRIMER PARAMETRO: referencia a variable tipo pthread_t del hilo
//SEGUNDO PARÁMETRO: (void *) &función_que_ejecuta_el_hilo()
//CUARTO PARÁMETRO: NULL porque la función no lleva parámetros.

pthread_create(&h2,NULL,(void *) &funcion_hilo2,&parametros_funcion)

pthread_join(h1,NULL);
pthread_join(h2,(void *) &retorno);

printf("",param);
}
```