

PThread Examples

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

void* do_loop (void* data)
{
    int i;
    int j;
    int me = *((int*)data);

    for (i=0; i<10; i++)
    {
        for (j=0; j<500000; j++) /*do nothing*/
            /* delay loop*/
            printf ("''%d' - Got '%d'\n", me, i);
    }

    /*terminate the thread*/
    pthread_exit(NULL);
}

int main (int argc, char* argv[])
{
    pthread_t p_thread;
    int thr_id;
    int a = 1;
    int b = 2;

    /*create a new thread to execute function do_loop*/
    thr_id = pthread_create (&p_thread, NULL, do_loop, (void*) &a);

    /*run do_loop in the main thread as well*/
    do_loop ((void*) &b);
}
```

```
    /*we should never reach here!*/  
    return 0;  
}
```

1. Note that `main` is also a thread.
2. The delay loop inside the function `do_loop` is used only to allow the two threads to interleave. If you have a fast CPU, you might need to use a longer delay to get interleaving.
3. Note that the `pthread_exit` call in the middle of the main program does not terminate main the way a `return` or `exit` would.

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

pthread_mutex_t mtx;

int data = 0;

void* proca()
{
    int a;

    pthread_mutex_lock(&mtx);
    a = data;
    a++;
    data = a;
    printf ("Data = '%d'\n", data);
    pthread_mutex_unlock(&mtx);
}

void* procb()
{
    int b;

    pthread_mutex_lock(&mtx);
    b = data;
    b--;
    data = b;
    printf ("Data = '%d'\n", data);
    pthread_mutex_unlock(&mtx);
}

int main()
{
5
```

```
pthread_t A, B;  
pthread_mutex_init(&mtx, NULL);  
  
pthread_create (&A, NULL, proca, NULL);  
pthread_create (&B, NULL, procb, NULL);  
  
pthread_exit(0);  
}
```