

3. Return Args

[threads-api](#)/thread_create_with_return_args.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <pthread.h>
4  #include "common\_threads.h"
5
6  typedef struct {
7      int a;
8      int b;
9  } myarg_t;
10
11 typedef struct {
12     int x;
13     int y;
14 } myret_t;
15
16 void *mythread(void *arg) {
17     myarg_t *args = (myarg_t *) arg;
18     printf("args %d %d\n", args->a, args->b);
19     myret_t *rvals = malloc(sizeof(myret_t));
20     assert(rvals != NULL);
21     rvals->x = 1;
22     rvals->y = 2;
23     return (void *) rvals;
24 }
25
26 int main(int argc, char *argv[]) {
27     pthread_t p;
28     myret_t *rvals;
29     myarg_t args = { 10, 20 };
30     Pthread_create(&p, NULL, mythread, &args);
31     Pthread_join(p, (void **) &rvals);
32     printf("returned %d %d\n", rvals->x, rvals->y);
33     free(rvals);
34     return 0;
35 }
```

```
// bad alternative:
void *mythread(void *arg) {
    myarg_t *args = (myarg_t *) arg;
    printf("args %d %d\n", args->a, args->b);
    myret_t oops = { 1, 2};
    return &oops; // mistake, local variable deallocated on return
}
```

```
$ gcc -Wall -I../include -o thread_create_with_return_args \
    thread_create_with_return_args.c -pthread
$ ./thread_create_with_return_args
args 10 20
returned 1 2
$
```