EXPERIMENT 4 – THREADS IN C

OBJECTIVE

Learn to threads for doing multiple function or parallel processing.

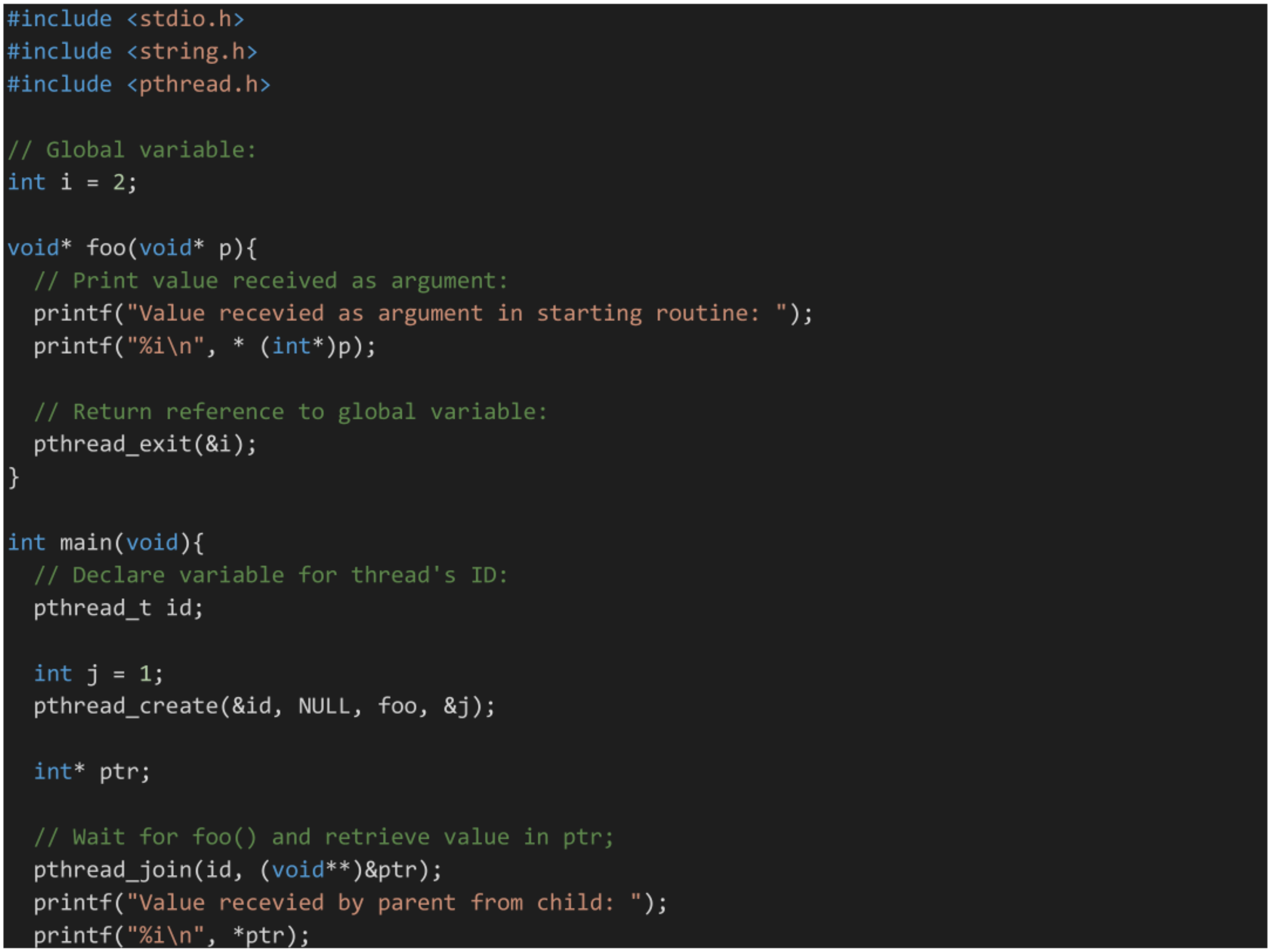
THREADS:

Application can have multiple processes and process have many threads. Thread is smallest unit of execution to which processor allocates time. It consists of:

- Program Counter (contains the address of next instruction to be executed)
- Stack
- Set of Registers
- Unique ID

However, a thread itself is not a program. It cannot run on its own but runs with in a program.

TASK 4.1: Create a Simple thread program:



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

// Global variable:
int i = 2;

void* foo(void* p){
  // Print value received as argument:
  printf("Value recevied as argument in starting routine: ");
  printf("%i\n", * (int*)p);

  // Return reference to global variable:
  pthread_exit(&i);
}

int main(void){
  // Declare variable for thread's ID:
  pthread_t id;

  int j = 1;
  pthread_create(&id, NULL, foo, &j);

  int* ptr;

  // Wait for foo() and retrieve value in ptr;
  pthread_join(id, (void**)&ptr);
  printf("Value recevied by parent from child: ");
  printf("%i\n", *ptr);
}
```

TASK 4.2:

How to write simple multi-threaded program in C.

```
#include <stdio.h>
#include<pthread.h>
void * show(void * u){
     printf(“new thread”);
}
int main(){
     pthread_t tid;
     pthread_create(&tid,NULL,&show,NULL);
     printf(“main thread”);
     pthread_join(tid,NULL);
     return 0;
}
```

For Compilation and execution:

```
gcc  -o out1 task1.c   –lpthread
./out1
```

Execute code and show outcome here:

Re-execute 3 more times and show outcomes. Did the outcomes change? Why?