# EXPERIMENT 4 – THREADS IN C

##### OBJECTIVE

Learn to threads for doing multiple function or parallel processing.

**TIME REQUIRED** : 3 hrs

**PROGRAMMING LANGUAGE** : C/C++/Java

**SOFTWARE REQUIRED** : Ubuntu/Fedora, gcc/gc, Windows, Dev, NetBeans

**HARDWARE REQUIRED** : Core i5 in Computer Labs

##### 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:

#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?

##### TASK 4.3:

#include <stdio.h> #include<pthread.h> pthread\_t thread[2];

static void \*funtion1(void \*\_){ int i;

for (i=0;i<=10;i++){ printf(“\nthread1 says%d”,i); sleep(1);

}

}

static void \*funtion2(void \*\_){ int i;

for (i=0;i<=5;i++){ printf(“\nthread2 says%d”,i); sleep(5);

}

}

int main(){ pthread\_create(&thread[0],NULL,function1,NULL); pthread\_create(&thread[1],NULL,function2,NULL); pthread\_exit(NULL);

return 1;

}

##### For Compilation and execution:

gcc -o out2 task2.c –lpthread

./out2

Execute Code and show outcome here: