Programming

- Create, compile and run thread1.c program
- Create, compile and run thread2.c program
- Create, compile and run thread3.c program
- Create, compile and run thread4.c program
- Create, compile and run thread5.c program

- Create, compile and run thread1.c program

```
babyearn@babyearn:~$ nano thread1.c
babyearn@babyearn:~$ more thread1.c
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

void *mythread (void *arg)
{
    printf(" message from thread: %s\n", (char*) arg);
    return NULL;
}

int main (int argc, char *argv[])
{
    pthread_t p1, p2;

    printf(" main program begins \n");

    pthread_create(&p1, NULL, mythread, "A");
    pthread_create(&p2, NULL, mythread, "B");

    pthread_join(p1, NULL);
    pthread_join(p2, NULL);

    printf(" main program ends \n");

    return 0;
}

babyearn@babyearn:~$ gcc -lpthread -o thread1 thread1.c
babyearn@babyearn:~$ ./thread1
 main program begins
 message from thread: A
 message from thread: B
 main program ends
babyearn@babyearn:~$
```

- Create, compile and run thread2.c program

```
babyearn@babyearn:~$ nano thread2.c
babyearn@babyearn:~$ more thread2.c
#include <assert.h>
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

typedef struct {
    int a;
    int b;
} myarg_t;

void *mythread (void *arg)
{
    myarg_t *args = (myarg_t *) arg;
    printf(" message from thread: %d and %d \n", args->a, args->b);
    return NULL;
}

int main (int argc, char *argv[])
{
    pthread_t p;
    myarg_t args = {10, 20};

    int rc = pthread_create(&p, NULL, mythread, &args);
    assert(rc == 0);

    pthread_join(p, NULL);
    printf(" main program done \n");
    return 0;
}
babyearn@babyearn:~$ gcc -lpthread -o thread2 thread2.c
babyearn@babyearn:~$ ./thread2
 message from thread: 10 and 20
 main program done
babyearn@babyearn:~$
```

- Create, compile and run thread3.c program

```
babyearn@babyearn:~$ nano thread3.c
babyearn@babyearn:~$ more thread3.c
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <unistd.h>

int sum;
void *runner(void *param);

int main(int argc, char *argv[])
{
    pthread_t tid;
    pthread_attr_t attr;

    int x = 5;
    sum = 0;

    pthread_attr_init(&attr);
    pthread_create(&tid, &attr, runner, (void *) &x);
    pthread_join(tid, NULL);

    printf(" sum = %d\n", sum);
    return 0;
}

void *runner(void *param)
{
    int i;
    int upper = *(int*) param;

    for (i = 1; i <= upper; i++)
        sum += i;
    pthread_exit(0);
}
babyearn@babyearn:~$ gcc -lpthread -o thread3 thread3.c
babyearn@babyearn:~$ ./thread3
 sum = 15
babyearn@babyearn:~$
```

Thread to do computation

The output is

```
babyearn@babyearn:~$ ./thread3
 sum = 15
```

- Create, compile and run thread4.c program

Thankamol Saekeang Sec:1 6687023

```
babyearn@babyearn:~$ nano thread4.c
babyearn@babyearn:~$ more thread4.c
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

void *mythread(void *arg)
{
    long long int value = (long long int) arg;
    printf(" thread received value = %lld\n", value);
    return (void*) (value + 1);
}

int main(int argc, char *argv[])
{
    pthread_t p;
    long long int rvalue;

    pthread_create(&p, NULL, mythread, (void*) 100);
    pthread_join(p, (void**) &rvalue);
    printf(" returned value = %lld\n", rvalue);

    return 0;
}
babyearn@babyearn:~$ gcc -lpthread -o thread4 thread4.c
babyearn@babyearn:~$ ./thread4
 thread received value = 100
 returned value = 101
babyearn@babyearn:~$
```

How to return a value from a thread?

```
pthread_join(p, (void**) &rvalue);
```

- Create, compile and run thread5.c program

```
babyearn@babyearn:~$ nano thread5.c
babyearn@babyearn:~$ more thread5.c
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

int x;
void *mythread (void *arg)
{
    x = 666;
    printf(" THE ANSWER IS = %d\n", *(int*) arg);
    return NULL;
}

int main(int argc, char *argv[])
{
    int z = 42;
    int k = 5;
    x = 0;
    pthread_t thread_ids[k];

    printf("[1]: x = %d\n", x);
    for (int i = 0; i < k; i++)
        pthread_create(&thread_ids[i], NULL, mythread, &z);

    for (int i = 0; i < k; i++)
        pthread_join(thread_ids[i], NULL);

    printf("[2]: x = %d\n", x);
    return 0;
}
babyearn@babyearn:~$ gcc -lpthread -o thread5 thread5.c
babyearn@babyearn:~$ ./thread5
[1]: x = 0
 THE ANSWER IS = 42
 THE ANSWER IS = 42
 THE ANSWER IS = 42
 THE ANSWER IS = 42
 THE ANSWER IS = 42
[2]: x = 666
babyearn@babyearn:~$
```