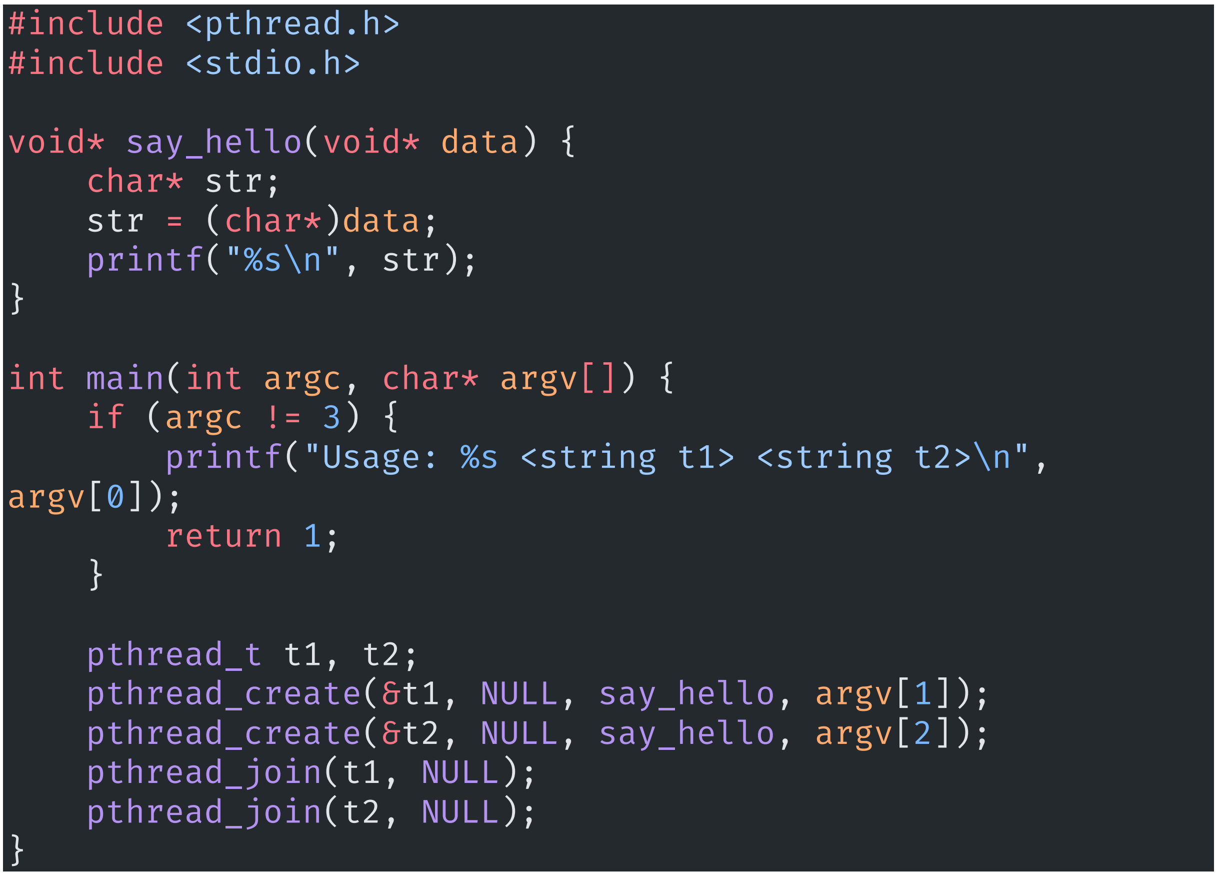
Activity 5 Pthread

MEMBERS:

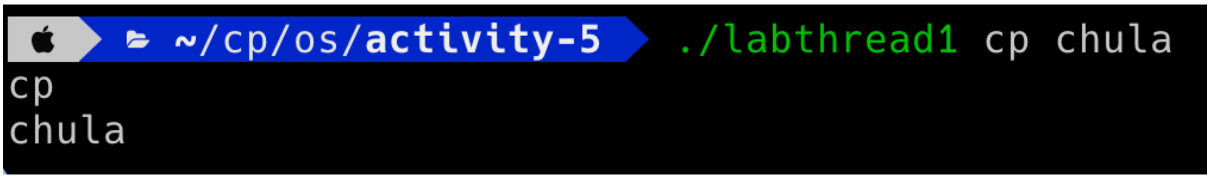
- 6531313221 Chotpisit Adunsehawat
- 6532040021 Chanatip Pattanapen
- 6532068721 Nutthapat Pongtanyavichai

Problem 1



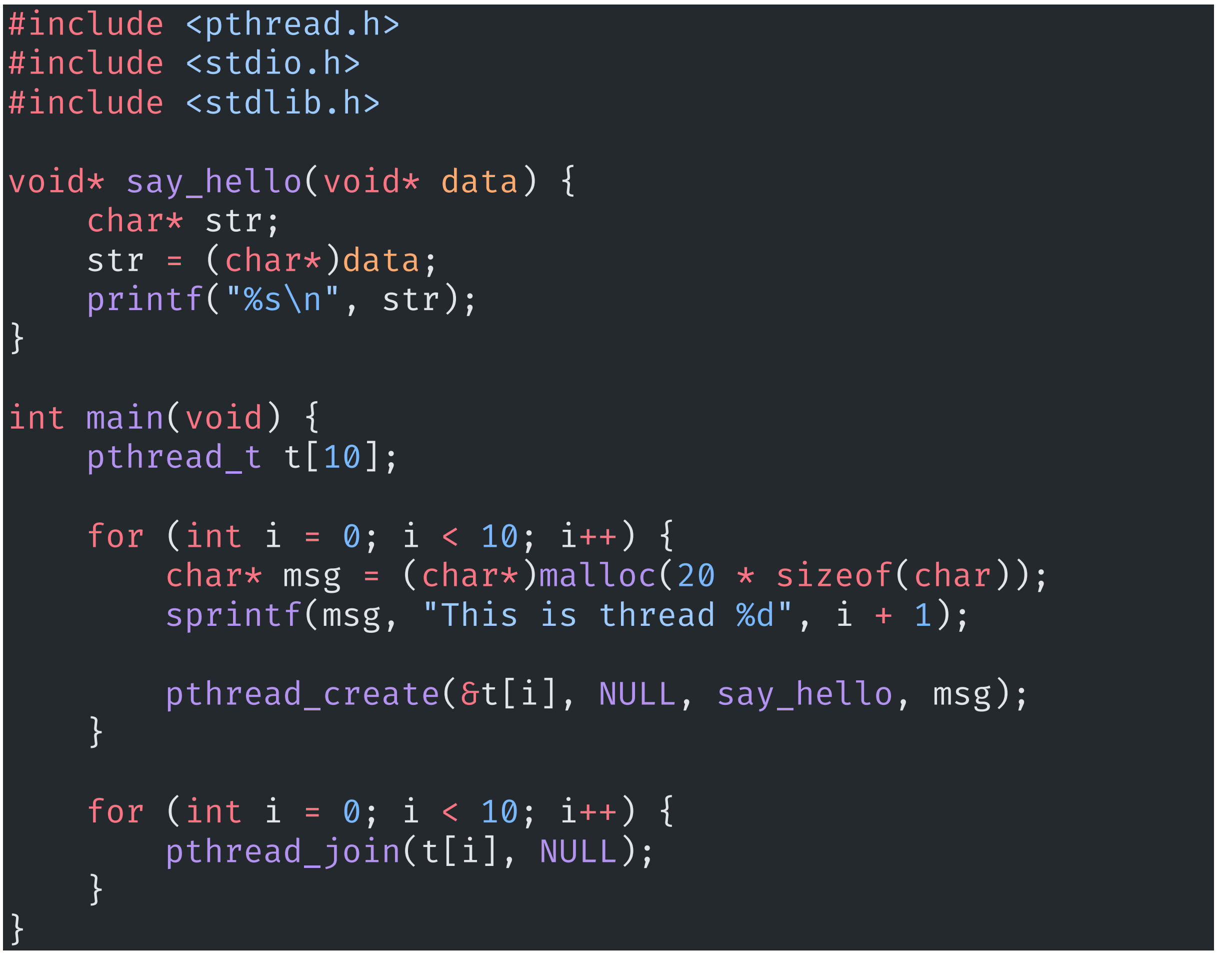
```c
#include <pthread.h>
#include <stdio.h>

void* say_hello(void* data) {
    char* str;
    str = (char*)data;
    printf("%s\n", str);
}

int main(int argc, char* argv[]) {
    if (argc != 3) {
        printf("Usage: %s <string t1> <string t2>\n",
argv[0]);
        return 1;
    }

    pthread_t t1, t2;
    pthread_create(&t1, NULL, say_hello, argv[1]);
    pthread_create(&t2, NULL, say_hello, argv[2]);
    pthread_join(t1, NULL);
    pthread_join(t2, NULL);
}
```

Result



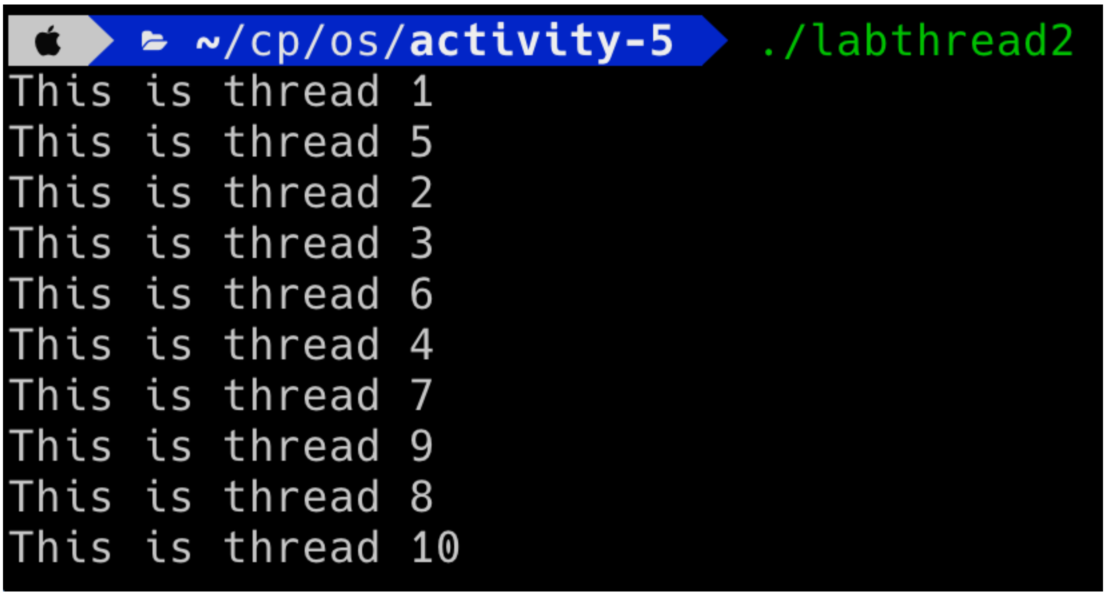
```
 ~/cp/os/activity-5  ./labthread1 cp chula
cp
chula
```

Problem 2



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

void* say_hello(void* data) {
    char* str;
    str = (char*)data;
    printf("%s\n", str);
}

int main(void) {
    pthread_t t[10];

    for (int i = 0; i < 10; i++) {
        char* msg = (char*)malloc(20 * sizeof(char));
        sprintf(msg, "This is thread %d", i + 1);

        pthread_create(&t[i], NULL, say_hello, msg);
    }

    for (int i = 0; i < 10; i++) {
        pthread_join(t[i], NULL);
    }
}
```

Result

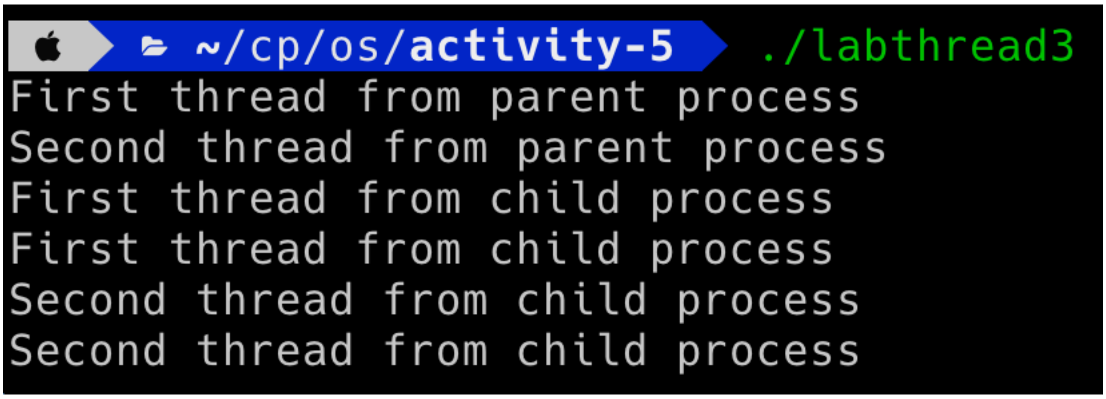


```
~/cp/os/activity-5  ./labthread2
This is thread 1
This is thread 5
This is thread 2
This is thread 3
This is thread 6
This is thread 4
This is thread 7
This is thread 9
This is thread 8
This is thread 10
```

Problem 3

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

void *child_thread(void *);
void *grandchild_thread(void *);

const int zero = 0;
const int one = 1;

int main(void) {
    pthread_t t1, t2;

    pthread_create(&t1, NULL, child_thread, &one);
    pthread_create(&t2, NULL, child_thread, &zero);

    pthread_join(t1, NULL);
    pthread_join(t2, NULL);
}

void *child_thread(void *isFirst) {
    if (*(int *)isFirst) {
        printf("First thread from parent process\n");
    } else {
        printf("Second thread from parent process\n");
    }

    pthread_t t1, t2;
    pthread_create(&t1, NULL, grandchild_thread, &one);
    pthread_create(&t2, NULL, grandchild_thread, &zero);

    pthread_join(t1, NULL);
    pthread_join(t2, NULL);
}

void *grandchild_thread(void *isFirst) {
    if (*(int *)isFirst) {
        printf("First thread from child process\n");
    } else {
        printf("Second thread from child process\n");
    }
}
```

Result



```
 ~/cp/os/activity-5  ./labthread3
First thread from parent process
Second thread from parent process
First thread from child process
First thread from child process
Second thread from child process
Second thread from child process
```