OPERATING SYSTEM LAB

Name: Thorat Amey Arun Reg No.: 23MCS1004

LAB EXPERIMENT 5
Threads

1. Create a thread pass string as your name to the thread and also print your registration number to main thread and print their IDs.

Program:

```
#include <stdio.h>
#include <pthread.h>
#include <unistd.h>
// Function to be executed in the thread
void* thread_function(void* arg) {
char* name = (char*)arg;
printf("Thread '%s' started\n", name);
printf("Thread ID: %ld\n", pthread_self());
// Thread logic goes here
return NULL;
}
int main() {
char rn[] = "23MCS1004";
printf("Main thread\n");
printf("Main thread ID: %ld\n",pthread_self());
printf("Registration Number: %s\n", rn);
char t[] = "THORAT AMEY ARUN";
pthread_t thread;
pthread_create(&thread, NULL, thread_function, (void*)t);
pthread_join(thread, NULL);
return 0;
}
```

Output:

```
vboxuser@Ubuntu:~/Desktop$ gcc p51.c -o p51 -lpthread
vboxuser@Ubuntu:~/Desktop$ ./p51
Main thread
Main thread ID: 139744399939392
Registration Number: 23MCS1004
Thread 'THORAT AMEY ARUN' started
Thread ID: 139744396703296
vboxuser@Ubuntu:~/Desktop$
```

OPERATING SYSTEM LAB

Name: Thorat Amey Arun Reg No.: 23MCS1004

2. Create two threads and display the two different function (Addition/ Odd or even ete.,) along with the corresponding thread_id. (pthread_self() function returns thread id)

Program:

```
#include <stdio.h>
#include <pthread.h>
void *addition_thread(void *arg) {
printf("Addition Thread: Thread ID = %ld\n", (long)pthread_self());
int result = 2 + 3;
printf("Addition Thread: Result = %d\n", result);
pthread_exit(NULL);
}
void *odd_even_thread(void *arg) {
printf("Odd Even Thread: Thread ID = %ld\n", (long)pthread_self());
int number = 7;
if (number % 2 == 0) {
printf("Odd/Even Thread: %d is even.\n", number);
} else {
printf("Odd/Even Thread: %d is odd.\n", number);
}
pthread_exit(NULL);
}
int main() {
pthread_t thread1, thread2;
pthread_create(&thread1, NULL, addition_thread, NULL);
pthread_create(&thread2, NULL, odd_even_thread, NULL);
pthread_join(thread1, NULL);
pthread_join(thread2, NULL);
return 0;
}
```

Output:

```
vboxuser@Ubuntu:~/Desktop$ gcc p52.c -o p52 -lpthread
vboxuser@Ubuntu:~/Desktop$ ./p52
Addition Thread: Thread ID = 140133747652160
Addition Thread: Result = 5
Odd Even Thread: Thread ID = 140133739259456
Odd/Even Thread: 7 is odd.
vboxuser@Ubuntu:~/Desktop$
```

OPERATING SYSTEM LAB

Name: Thorat Amey Arun Reg No.: 23MCS1004

3. Design two threads to count the vowels and consonants either from text file or a given string.

Program:

```
#include <stdio.h>
#include <pthread.h>
#include <ctype.h>
int vowelCount = 0;
int consonantCount = 0;
void *countVowels(void *arg) {
char *text = (char *)arg;
for (int i = 0; text[i] != '\0'; i++) {
char c = tolower(text[i]);
if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {
vowelCount++;
}}
pthread_exit(NULL);
}
void *countConsonants(void *arg) {
char *text = (char *)arg;
for (int i = 0; text[i] != '\0'; i++) {
char c = tolower(text[i]);
if (isalpha(c) && c != 'a' && c != 'e' && c != 'i' && c != 'o' && c != 'u') {
consonantCount++;
}
}
pthread_exit(NULL);
}
int main() {
pthread_t vowelThread, consonantThread;
char text[] = "THORAT AMEY ARUN";
pthread_create(&vowelThread, NULL, countVowels, (void *)text);
pthread_create(&consonantThread, NULL, countConsonants, (void
*)text);
pthread_join(vowelThread, NULL);
pthread_join(consonantThread, NULL);
printf("Vowel Count: %d\n", vowelCount);
printf("Consonant Count: %d\n", consonantCount);
return 0;
}
```

OPERATING SYSTEM LAB

Name: Thorat Amey Arun Reg No.: 23MCS1004

Output:

```
vboxuser@Ubuntu:~/Desktop$ gcc p53.c -o p53 -lpthread
vboxuser@Ubuntu:~/Desktop$ ./p53
Vowel Count: 6
Consonant Count: 8
vboxuser@Ubuntu:~/Desktop$
```

4. Write a C program to create a thread to do the following. Thread performs payroll calculation of employee by using a formula

   Salary = Basic + 25% of Basic as HRA+ 40% of basic as DA+ Incentive.

   Calculate the salary of 3 employees and display the name and their salary in main().

   Provide the required input.

   Program:

```
#include <stdio.h>
#include <pthread.h>
// Structure to hold employee information
struct Employee {
char name[50];
double basic;
double incentive;
double salary;
};
// Function executed by the thread
void *calculateSalary(void *arg) {
struct Employee *employee = (struct Employee *)arg;
employee->salary = employee->basic + 0.25 * employee->basic + 0.4 *
employee->basic + employee->incentive;
pthread_exit(NULL);
}
int main() {
pthread_t thread[3];
struct Employee employees[3];
// Input employee information
for (int i = 0; i < 3; i++) {
printf("Enter details for Employee %d:\n", i + 1);
printf("Name: ");
```

```
scanf("%s", employees[i].name);
printf("Basic Salary: ");
scanf("%lf", &employees[i].basic);
printf("Incentive: ");
scanf("%lf", &employees[i].incentive);
}
// Create threads to calculate salaries
for (int i = 0; i < 3; i++) {
pthread_create(&thread[i], NULL, calculateSalary, (void
*)&employees[i]);
}
// Wait for threads to finish
for (int i = 0; i < 3; i++) {
pthread_join(thread[i], NULL);
}
// Display employee salaries
printf("\nEmployee Salaries:\n");
for (int i = 0; i < 3; i++) {
printf("Name: %s, Salary: %.2lf\n", employees[i].name,
employees[i].salary);
}
return 0;
}
```

Output:

```
vboxuser@Ubuntu:~/Desktop$ gcc p54.c -o p54 -lpthread
vboxuser@Ubuntu:~/Desktop$ ./p54
Enter details for Employee 1:
Name: Amey
Basic Salary: 25000
Incentive: 200
Enter details for Employee 2:
Name: Avdhut
Basic Salary: 20000
Incentive: 300
Enter details for Employee 3:
Name: Anita
Basic Salary: 30000
Incentive: 250

Employee Salaries:
Name: Amey, Salary: 41450.00
Name: Avdhut, Salary: 33300.00
Name: Anita, Salary: 49750.00
vboxuser@Ubuntu:~/Desktop$
```