

LAB 5

5) Write a C program to simulate the dining philosopher's problem.

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

#define N 5
#define THINKING 0
#define HUNGRY 1
#define EATING 2

int state[N] = {0};
int forks[N] = {1, 1, 1, 1, 1};
int hungry[N] = {0};

void wait(int *S) {
    while (*S <= 0);
    (*S)--;
}

void signal(int *S) {
    (*S)++;
}

int left(int id) {
    return id;
}

int right(int id) {
    return (id + 1) % N;
}

void simulate(int max_eaters) {
    int done = 0;

    while (!done) {
        int eating_count = 0;
        done = 1;

        for (int i = 0; i < N; i++) {
            if (hungry[i]) {
                printf("P %d is waiting\n", i + 1);
                done = 0;
            }
        }
    }
}
```

```

        for (int i = 0; i < N && eating_count < max_eaters; i++) {
            if (hungry[i]) {
                if (forks[left(i)] == 1 && forks[right(i)] == 1) {
                    wait(&forks[left(i)]);
                    wait(&forks[right(i)]);

                    printf("P %d is granted to eat\n", i + 1);
                    state[i] = EATING;
                    printf("P %d has finished eating\n", i + 1);
                    state[i] = THINKING;

                    signal(&forks[left(i)]);
                    signal(&forks[right(i)]);

                    hungry[i] = 0;
                    eating_count++;
                }
            }
        }
    }
}

int main() {
    int total, hungry_count, id, choice;

    printf("Enter the total number of philosophers: ");
    scanf("%d", &total);

    printf("How many are hungry: ");
    scanf("%d", &hungry_count);

    for (int i = 0; i < hungry_count; i++) {
        printf("Enter philosopher %d position (1 to %d): ", i + 1, total);
        scanf("%d", &id);
        hungry[id - 1] = 1;
        state[id - 1] = HUNGRY;
    }

    while (1) {
        printf("\n1. One can eat at a time\n");
        printf("2. Two can eat at a time\n");
        printf("3. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        if (choice == 3) break;
    }
}

```

```

        if (choice == 1 || choice == 2) {
            simulate(choice);
        } else {
            printf("Invalid choice!\n");
        }
    }

    return 0;
}

```

OUTPUT:

```

PS D:\002> cd "d:\002\" ; if ($?) { gcc lab5.c -o lab5 } ; if ($?) { .\lab5 }
Enter the total number of philosophers: 5
How many are hungry: 3
Enter philosopher 1 position (1 to 5): 2
Enter philosopher 2 position (1 to 5): 4
Enter philosopher 3 position (1 to 5): 5

1. One can eat at a time
2. Two can eat at a time
3. Exit
Enter your choice: 1
P 2 is waiting
P 4 is waiting
P 5 is waiting
P 2 is granted to eat
P 2 has finished eating
P 4 is waiting
P 5 is waiting
P 4 is granted to eat
P 4 has finished eating
P 5 is waiting
P 5 is granted to eat
P 5 has finished eating

1. One can eat at a time
2. Two can eat at a time
3. Exit
Enter your choice: █

```