Operating Systems
CT-353

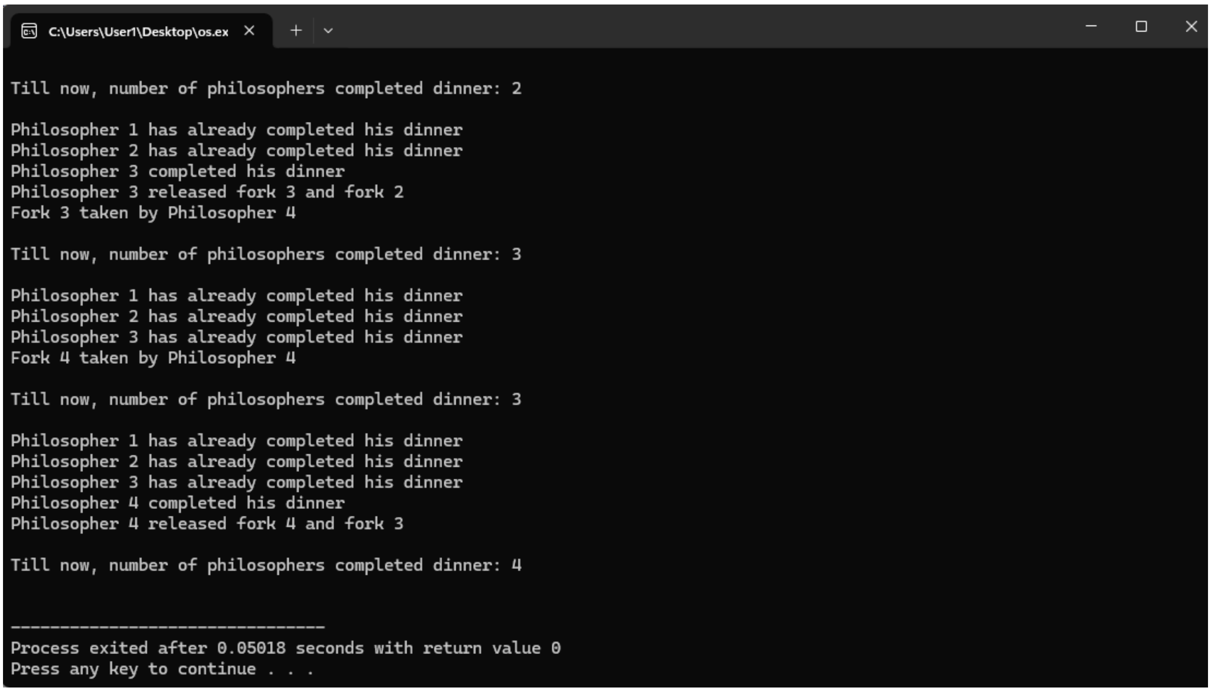
Name: Sabrina Shahzad
Roll No. : DT-026

Lab 06:
Dining Philosophers Problem

```
#include <stdio.h>

#define n 4

int compltedPhilo = 0, i;

struct fork {
    int taken;
} ForkAvil[n];

struct philosp {
    int left;
    int right;
} Philostatus[n];

void goForDinner(int philID) {
    // If philosopher has completed dinner
    if (Philostatus[philID].left == 10 && Philostatus[philID].right == 10) {
        printf("Philosopher %d has already completed his dinner\n", philID + 1);
    }

    // If philosopher has both forks
    else if (Philostatus[philID].left == 1 && Philostatus[philID].right == 1) {
        printf("Philosopher %d completed his dinner\n", philID + 1);

        Philostatus[philID].left = Philostatus[philID].right = 10; // Mark as finished

        int otherFork = philID - 1;
        if (otherFork == -1)
            otherFork = n - 1;

        ForkAvil[philID].taken = ForkAvil[otherFork].taken = 0; // release forks
        printf("Philosopher %d released fork %d and fork %d\n", philID + 1, philID + 1,
otherFork + 1);
        compltedPhilo++;
    }

    // Has left fork, trying for right
    else if (Philostatus[philID].left == 1 && Philostatus[philID].right == 0) {
        if (philID == (n - 1)) { // Last philosopher
            if (ForkAvil[philID].taken == 0) {
                ForkAvil[philID].taken = Philostatus[philID].right = 1;
                printf("Fork %d taken by Philosopher %d\n", philID + 1, philID + 1);
            } else {
                printf("Philosopher %d is waiting for fork %d\n", philID + 1, philID + 1);
            }
        } else {
            int dupphilID = philID;
            philID -= 1;
            if (philID == -1)
                philID = n - 1;

            if (ForkAvil[philID].taken == 0) {
                ForkAvil[philID].taken = Philostatus[dupphilID].right = 1;
                printf("Fork %d taken by Philosopher %d\n", philID + 1, dupphilID + 1);
            } else {
                printf("Philosopher %d is waiting for Fork %d\n", dupphilID + 1, philID + 1);
            }
        }
    }

    // Has not taken any fork yet
    else if (Philostatus[philID].left == 0) {
        if (philID == (n - 1)) {
            if (ForkAvil[philID - 1].taken == 0) {
                ForkAvil[philID - 1].taken = Philostatus[philID].left = 1;
                printf("Fork %d taken by Philosopher %d\n", philID, philID + 1);
            } else {
                printf("Philosopher %d is waiting for fork %d\n", philID + 1, philID);
            }
        } else {
            if (ForkAvil[philID].taken == 0) {
                ForkAvil[philID].taken = Philostatus[philID].left = 1;
                printf("Fork %d taken by Philosopher %d\n", philID + 1, philID + 1);
            } else {
                printf("Philosopher %d is waiting for fork %d\n", philID + 1, philID + 1);
            }
        }
    }
```

```
}

int main() {
    for (i = 0; i < n; i++) {
        ForkAvil[i].taken = 0;
        Philostatus[i].left = 0;
        Philostatus[i].right = 0;
    }

    while (compltedPhilo < n) {
        for (i = 0; i < n; i++)
            goForDinner(i);

        printf("\nTill now, number of philosophers completed dinner: %d\n\n", compltedPhilo);
    }

    return 0;
}
```

Output:



```
Till now, number of philosophers completed dinner: 2

Philosopher 1 has already completed his dinner
Philosopher 2 has already completed his dinner
Philosopher 3 completed his dinner
Philosopher 3 released fork 3 and fork 2
Fork 3 taken by Philosopher 4

Till now, number of philosophers completed dinner: 3

Philosopher 1 has already completed his dinner
Philosopher 2 has already completed his dinner
Philosopher 3 has already completed his dinner
Fork 4 taken by Philosopher 4

Till now, number of philosophers completed dinner: 3

Philosopher 1 has already completed his dinner
Philosopher 2 has already completed his dinner
Philosopher 3 has already completed his dinner
Philosopher 4 completed his dinner
Philosopher 4 released fork 4 and fork 3

Till now, number of philosophers completed dinner: 4


--------------------------------
Process exited after 0.05018 seconds with return value 0
Press any key to continue . . .
```