National University of Computer and Emerging Sciences

Laboratory Manual

for

Computer Organization and Assembly Language Programming

Lab Instructor	Sana Ejaz
Semester	Fall 2024

Department of Computer Science

FAST-NU, Lahore, Pakistan

OBJECTIVES:

- Understand and implement timer-based scheduling for multitasking, allowing multiple processes to execute concurrently.

- Expand an existing timer scheduling code to handle additional processes and modify their respective tasks.
- Learn to create and manage threads using keyboard-driven multitasking and dynamic thread initialization.

Task 1: Assemble and run the code given in (example 10.1 hardcopy/ 11.1 softcopy):

Task 2: In above code, timer schedules following 3 processes:

Process 0: jmp $
Process 1: Printing rotation on location [0][0] (task one)
Process 2: Printing rotation on location [0][79] (task two)

Update this program such that time schedules 5 processes:

Process 0: jmp $
Process 1: Printing rotation on location [0][0] (task one)
Process 2: Printing rotation on location [0][79] (task two)
Process 3: Printing rotation on location [20][0] (task three)
Process 4: Printing rotation on location [20][79] (task four)

Task 3: Assemble and run the code in (example 11.2), it takes a key from user, upon getting key it starts a new thread of infinite number printing at next line (column 70).