

Lab Exercise - 9

❖ AIM :: WAP to perform Priority Scheduling.

Source_Code ::

```
echo $'\n' "5C6 - Amit Singhal (11614802722)" $'\n'
```

```
# Read the number of processes
```

```
read -p "Enter the number of processes: " num_processes
```

```
echo $'\n'
```

```
# Declare arrays for storing process information
```

```
declare -a arrival
```

```
declare -a burst
```

```
declare -a priority
```

```
declare -a completion
```

```
declare -a waiting
```

```
declare -a turnaround
```

```
declare -a process_ids
```

```
declare -a remaining_burst
```

```
# Input arrival time, burst time, and priority for each process
```

```
for ((i=0; i<num_processes; i++))
```

```
do
```

```
    process_ids[$i]=$((i+1))
```

```
    echo -n "Enter Arrival Time, Burst Time, and Priority for Process $((i+1)): "
```

```
    read arrival[$i] burst[$i] priority[$i]
```

```
    remaining_burst[$i]={burst[$i]} # Initialize remaining burst time
```

```
    completion[$i]=0 # Initialize completion time to 0
```

```
done
```

```
# Priority scheduling with preemption
```

```
priority_scheduling() {
```

```
    time=0
```

```
    completed=0
```

```
    gantt_chart=""
```

```
    prev_process=-1
```

```

while [ $completed -lt $num_processes ]; do
    # Find the process with the highest priority that has arrived and has remaining burst time
    highest_priority=-1
    current_process=-1

    for ((i=0; i<num_processes; i++)); do
        if [ ${arrival[$i]} -le $time ] && [ ${remaining_burst[$i]} -gt 0 ]; then
            if [ $highest_priority -eq -1 ] || [ ${priority[$i]} -lt $highest_priority ]; then
                highest_priority=${priority[$i]}
                current_process=$i
            fi
        fi
    done

    if [ $current_process -ne -1 ]; then
        if [ $current_process -ne $prev_process ]; then
            gantt_chart+=" $time -- P${process_ids[$current_process]} -- "
        fi

        remaining_burst[$current_process]=$((remaining_burst[$current_process] - 1))
        time=$((time + 1))

        # If the process finishes, calculate its completion, turnaround, and waiting times
        if [ ${remaining_burst[$current_process]} -eq 0 ]; then
            completion[$current_process]=$time
            turnaround[$current_process]=$((completion[$current_process] -
                                                    arrival[$current_process]))
            waiting[$current_process]=$((turnaround[$current_process] -
                                                    burst[$current_process]))

            completed=$((completed + 1))
        fi

        prev_process=$current_process
    else
        gantt_chart+=" $time -- XX -- "
        time=$((time + 1))
    fi
done
gantt_chart+=" $time" # Add the final time to Gantt chart
}

```

```
# Function to display the Gantt chart
```

```
display_gantt_chart() {  
    echo $'\n'"Gantt Chart:"  
    echo "$gantt_chart"  
}
```

```
# Function to display the process table with calculated times
```

```
display_results() {  
    echo $'\n'"PID | AT | BT | Priority | CT | TAT | WT |"  
    echo "-----"  
    for ((i=0; i<num_processes; i++)); do  
        printf "P%-3d | %-3d | %-2d |   %-4d | %-3d | %-3d | %-3d |\n" \  
            "${process_ids[$i]}" "${arrival[$i]}" "${burst[$i]}" "${priority[$i]}" \  
            "${completion[$i]}" "${turnaround[$i]}" "${waiting[$i]}"  
    done  
    echo "-----"  
}
```

```
# Function to calculate and display the average waiting and turnaround times
```

```
calculate_averages() {  
    total_waiting=0  
    total_turnaround=0  
  
    for ((i=0; i<num_processes; i++)); do  
        total_waiting=$((total_waiting + waiting[$i]))  
        total_turnaround=$((total_turnaround + turnaround[$i]))  
    done  
  
    avg_waiting=$(echo "scale=2; $total_waiting / $num_processes" | bc)  
    avg_turnaround=$(echo "scale=2; $total_turnaround / $num_processes" | bc)  
  
    echo $'\n'"Average Waiting Time <WT> :: $avg_waiting"  
    echo "Average Turnaround Time <TAT> :: $avg_turnaround"  
}
```

```
# Run the priority scheduling algorithm with preemption
```

```
priority_scheduling
```

```
# Display the Gantt chart
```

```
display_gantt_chart
```

```
# Display the process table
```

```
display_results
```

```
# Calculate and display the averages
```

```
calculate_averages
```

Output ::

```
singhal-amit@singhal-amit-ThinkPad-T430:~/Downloads$ chmod +x prg9
singhal-amit@singhal-amit-ThinkPad-T430:~/Downloads$ ./prg9
```

```
5C6 - Amit Singhal (11614802722)
```

```
Enter the number of processes: 4
```

```
Enter Arrival Time, Burst Time, and Priority for Process 1: 0 4 2
```

```
Enter Arrival Time, Burst Time, and Priority for Process 2: 1 3 1
```

```
Enter Arrival Time, Burst Time, and Priority for Process 3: 2 5 3
```

```
Enter Arrival Time, Burst Time, and Priority for Process 4: 3 2 4
```

```
Gantt Chart:
```

```
0 -- P1 -- 1 -- P2 -- 4 -- P1 -- 7 -- P3 -- 12 -- P4 -- 14
```

PID	AT	BT	Priority	CT	TAT	WT
P1	0	4	2	7	7	3
P2	1	3	1	4	3	0
P3	2	5	3	12	10	5
P4	3	2	4	14	11	9

```
Average Waiting Time <WT> :: 4.25
```

```
Average Turnaround Time <TAT> :: 7.75
```