**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 ::