



L OVELY
P ROFESSIONAL
U NIVERSITY

OPERATING SYSTEMS PROJECT REPORT

QUESTION NO 14

Write a program to implement priority scheduling algorithm with context switching time . Prompt to user to enter the number of processes and then enter their priority, burst time and arrival time also. Now whenever operating system preempts a process and shifts cpu's control to some another process of higher priority assume that it takes 2 seconds for context switching (dispatcher latency).Form a scenario , where we can give the processes are assigned with priority where the lower integer number is higher priority and then context switch .. as the process waits the priority of the process increase at rate of one per 2 time units of wait. Calculate waiting time and turnaround time for each process.

SUBMITTED BY:

SANIDHYA DASH

Reg No. - 11802151

Roll – 39

Section – K18GA

SUBMITTED TO:

NAVNEET KAUR

Description

To implement priority scheduling algorithm with context switching time. Prompt to user to enter the number of processes and then enter their priority, burst time and arrival time also. Now whenever operating system preempts a process and shifts CPU's control to some another process of higher priority assume that it takes 2 seconds for context switching(dispatcher latency).Form a scenario, where we can give the processes are assigned with priority where the lower integer number is higher priority and then context switch as the process waits the priority of the process increase at rate of one per 2-time units of wait. Calculate waiting time and turnaround time for each process.

Algorithm

Step 1: Taking Input 'n' for no. of processes

Step 2: Run LOOP till ($I < N$)

Taking Input 'arrivaltime', 'bursttime', 'priority' for Arrival time, Burst time and Priority of the Processes.

Step 3: Run LOOP till ($I < N$)

Set PTR=I;

Run LOOP till ($J < N$) from $J = I + 1$

IF(Priority[I] < Priority[PTR])

Do PTR=J

SET Temp=Priority[I];

SET Priority[I]=Priority[PTR];

SET Priority[PTR]=Temp;

SET Temp=BurstTime[I];

SET BurstTime[I]=BurstTime[PTR];

SET BurstTime[PTR]=Temp;

SET Temp=PNo[I];

SET PNo[I]=PNo[PTR];

SET PNo[PTR]=Temp;

Step 4: SET WaitingTime[0]=0

Run LOOP till ($I < N$)

SET WaitingTime[I]=0;

Run LOOP till($J < I$)

SET WaitingTime[I] = WaitingTime[I] + BurstTime[J];

[END LOOP]

SET Sum=Sum+WaitingTime[I];

[END LOOP]

Step 5: SET AvgWaitTime= Sum/N;

Step 6: Run LOOP till(I<N)

SET TurnAroundTime[I]=BurstTime[I]+WaitingTime[I]

SET Sum1=Sum1+TurnAroundTime[I];

Step 7: SET AvgTurnAroundTime=Sum1/n;

[EXIT]

CODE

// CODE BY SANIDHYA DASH - CSE316 - OS PROJECT

// FOR MORE INFORMATION CONTACT: sanidhyadash32@gmail.com

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

```
int main()
```

```
{
```

```
    char ch;
```

```
    int n,i,j,ptr,c,z,sum=0,avgttime;
```

```
    int arrivaltime[10],bursttime[10],waitingtime[10],priority[10],pn[10],tetime[10];
```

```
    printf("Enter the the no. of processes : ");
```

```
    scanf("%03d",&n);
```

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

```
    {
```

```
        printf("\n\nprocess [%03d] \n",i+1);
```

```
        printf("Enter arrival time : ");
```

```
        scanf("%03d",&arrivaltime[i]);
```

```
    printf("Enter Burst time : ");
```

```
    fflush(stdin);
```

```
        scanf("%03d",&bursttime[i]);
```

```
        printf("Enter Priority : ");
```

```
        fflush(stdin);
```

```
        scanf("%03d",&priority[i]);
```

```
        pn[i]=i+1;
```

```
    }
```

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

```
    {
```



```
ptr=i;

for (j=i+1;j<n;j++)
{
    if(priority[j]<priority[ptr])
    {
        ptr=j;
    }
}
c=priority[i];
priority[i]=priority[ptr];
priority[ptr]=c;

c=bursttime[i];
bursttime[i]=bursttime[ptr];
bursttime[ptr]=c;
c=pn[i];
pn[i]=pn[ptr];
pn[ptr]=c;

}
waitingtime[0]=0;
for(i=1;i<n;i++)
{
    waitingtime[i]=0;
    for(j=0;j<i;j++)
    {
        waitingtime[i] = waitingtime[i]+bursttime[j];
    }

    z = z+waitingtime[i];

}
avgtime=z/n;
```

```
fflush(stdin);

printf("| PROCESS | ARRIVAL TIME | BURST TIME | WAITING TIME |
PRIORITY | TURN AROUND TIME |\n");

for (i=0;i<n;i++)
{
    ttime[i]=bursttime[i]+waitingtime[i];
    printf("| %03d | %03d | %03d | %03d | %03d |
%03d |\n",pn[i],arrivaltime[i],bursttime[i],waitingtime,pn[i],ttime[i]);
    sum= sum+ttime[i];

}

int avgtatime= sum/n;

printf("\n| AVERAGE WAIT TIME : %03d |",avgtime);
printf("\n| AVERAGE TURNAROUND TIME : %03d |",avgtatime);

}
```

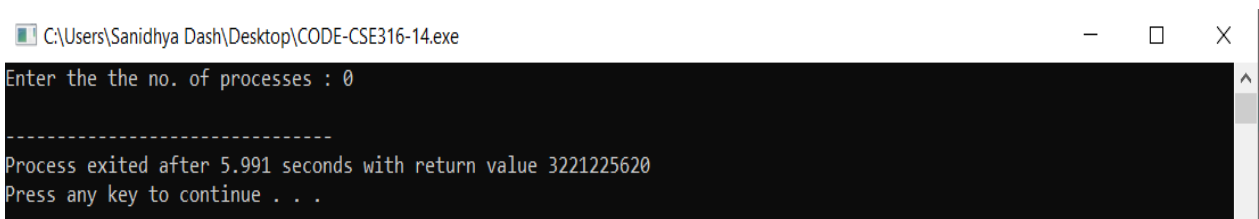
BOUNDARY CONDITIONS

- User **cannot** pass process value **greater than 10**.
- Arrival time value is bounded for **10 processes**.
- Burst time value is bounded for **10 processes**.
- Priority of processes cannot exceed the **limit 10**.
- Buffer memory may exist, to rectify it **fflush(stdin)** function is used.

TEST CASES

Test case 1: When User Pass the value of processes = 0

Result: Test Case Passed



```
C:\Users\Sanidhya Dash\Desktop\CODE-CSE316-14.exe
Enter the the no. of processes : 0
-----
Process exited after 5.991 seconds with return value 3221225620
Press any key to continue . . .
```

Test case 2: When User Pass the value of processes < 0

Result: Test Case partially Passed

```
C:\Users\Sanidhya Dash\Desktop\CODE-CSE316-14.exe
Enter the the no. of processes : -8
|  PROCESS  |  ARRAIVAL TIME  |  BURST TIME  |  WAITING TIME  |  PRIORITY  |  TURN AROUND TIME  |
|  AVERAGE WAIT TIME      : -06  |
|  AVERAGE TURNAROUND TIME : 000  |
-----
Process exited after 5.944 seconds with return value 0
Press any key to continue . . .
```

Test case 3: When User pass the value of processes > 0

Result: Test Case Passed

```
C:\Users\Sanidhya Dash\Desktop\CODE-CSE316-14.exe
Enter the the no. of processes : 2

process [001]
Enter arrival time : 50
50
Enter Burst time : 50
50
Enter Priority : 50
50

process [002]
Enter arrival time : 30
Enter Burst time : 20
20
Enter Priority : 10
10
|  PROCESS  |  ARRAIVAL TIME  |  BURST TIME  |  WAITING TIME  |  PRIORITY  |  TURN AROUND TIME  |
|  002      |  050           |  020         |  6487360       |  002       |  020              |
|  001      |  050           |  050         |  6487360       |  001       |  070              |
|  AVERAGE WAIT TIME      : 035  |
|  AVERAGE TURNAROUND TIME : 045  |
-----
Process exited after 70.87 seconds with return value 0
Press any key to continue . . .
```

Test case 4: When User give input = 0, for arrival time, burst time and priority.

Result: Test Case Passed with all desired output, i.e=0

C:\Users\Sanidhya Dash\Desktop\CODE-CSE316-14.exe

Enter the the no. of processes : 3

process [001]

Enter arrival time : 00

00

Enter Burst time : 00

00

Enter Priority : 00

00

process [002]

Enter arrival time : 00

Enter Burst time : 00

00

Enter Priority : 00

00

process [003]

Enter arrival time : 00

Enter Burst time : 00

00

Enter Priority : 00

00

PROCESS	ARRAIVAL TIME	BURST TIME	WAITING TIME	PRIORITY	TURN AROUND TIME
001	000	000	6487360	001	000
002	000	000	6487360	002	000
003	000	000	6487360	003	000

AVERAGE WAIT TIME : 016

AVERAGE TURNAROUND TIME : 000

Process exited after 21.57 seconds with return value 0

Press any key to continue . . .

Test case 5: When user gives the process value > 10 i.e bounded length for input.

Status: exceed the input boundary limit, giving output with garbage values. (Passed)

C:\Users\Sanidhya Dash\Desktop\CODE-CSE316-14.exe

process [011]

Enter arrival time : 11

Enter Burst time : 2

2

Enter Priority : 11

11

process [012]

Enter arrival time : 12

Enter Burst time : 1

1

Enter Priority : 12

12

PROCESS	ARRAIVAL TIME	BURST TIME	WAITING TIME	PRIORITY	TURN AROUND TIME
001	001	012	6487360	001	012
002	001	011	6487360	002	023
003	002	010	6487360	003	033
004	003	009	6487360	004	042
005	004	008	6487360	005	050
006	005	007	6487360	006	057
007	006	006	6487360	007	063
008	007	005	6487360	008	068
009	008	004	6487360	009	072
010	009	003	6487360	010	075
011	010	002	6487360	011	077
012	011	001	6487360	012	078
012	012	000	6487360	012	001
023	011	012	6487360	023	014
033	010	023	6487360	033	026
042	009	033	6487360	042	037
050	008	042	6487360	050	047
057	007	050	6487360	057	056
063	006	057	6487360	063	064
000	078	000	6487360	000	000
-17254715	001	000	6487360	-17254715	000
127	014	000	6487360	127	000
000	026	050	6487360	000	050
000	291	000	6487360	000	000
000	047	4229664	6487360	000	4229664
000	056	000	6487360	000	000

AVERAGE WAIT TIME : 014

AVERAGE TURNAROUND TIME : 54230

Process exited after 714.3 seconds with return value 0

Press any key to continue . . .



Test case 6: When user gives all the desired input. i.e no. of process > 0, burst time>0, arrival time>0, priority >=0;

Status: All test case Passed

```
C:\Users\Sanidhya Dash\Desktop\CODE-CSE316-14.exe
Enter the the no. of processes : 2

process [001]
Enter arrival time : 40
40
Enter Burst time : 50
50
Enter Priority : 7
7

process [002]
Enter arrival time : 20
Enter Burst time : 90
90
Enter Priority : 2
2
|  PROCESS   |  ARRAIVAL TIME  |  BURST TIME  |  WAITING TIME  |  PRIORITY  |  TURN AROUND TIME  |
|  002      |  040           |  090         |  6487360      |  002       |  090               |
|  001      |  007           |  050         |  6487360      |  001       |  140               |
|  AVERAGE WAIT TIME      : 070 |
|  AVERAGE TURNAROUND TIME : 115 |
-----
Process exited after 39.57 seconds with return value 0
Press any key to continue . . .
```