OPERATING SYSTEM PROJECT  
  
  
REAL TIME SCHEDULING

Talha Asif k142246

Syed Muhammad Abdurrehman k142429

TASK: Real time Scheduling Algorithms.

ABSTRACT:

Scheduling algorithms are a leading part of real-time systems and there exists many different scheduling algorithms due to the varying needs and requirements of different real time systems. The choice of algorithm is important in every real-time system and is greatly influenced by what kind of system the algorithm will serve.

INTRODUCTION:

There are many Real time scheduling algorithms. This report refer to the functioning of two real time scheduling algorithms i.e Rate Monotonic and Earliest Job first scheduling algorithm. Both the algorithms have their unique significance in Real time Scheduling and Real time Systems. Rate monotonic work on priority base while Earliest job first algorithm work on Deadline base.

WORK FORUM:

We have coded algorithms in C language on Linux. Processes are input by filing and their output is also saved into to a particular file.

MEHTODOLOGY:

We have used basic methodology for this project.  
  
Rate Monotonic.

A1/b1 + A2/b2 …. An/bn < n \* (2^1/n -1) - if this condition satisfies then we are able to schedule it by rate monotonic else we cannot.  
We have created here our own priority by considering process with smallest burst time has highest priority.  
  
 Earliest Job First.  
A1/b1 + A2/b2 …. An/bn < 1 - if this condition satisfies then we are able to schedule these processes by EDF scheduling.  
Here the process with earliest deadline get the processor first.

Input is done with file while output is saved in file and showed on screen both.  
  
to compile and run.

gcc –o p1 project.c –lm  
 chmod a+x  
 ./p1

CODE:

Code how we sort process according to Earliest deadline and then schedule it linearly.

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

{

for(j=0;j<n-i;j++)

{

if(dead[j+1]<dead[j])

{

temp = dead[j];

dead[j]=dead[j+1];

dead[j+1]=temp;

temp = burst[j];

burst[j]=burst[j+1];

burst[j+1]=temp;

temp = pro[j];

pro[j]=pro[j+1];

pro[j+1]=temp;

}

}

}

To calculate waiting and turnaround time for processes

for(i=2;i<=n;i++)

{

turn[i] = burst[i]+turn[i-1];

wt[i] = turn[i]-burst[i];

}

for(i=2;i<=n;i++)

{

avgwt = avgwt + (float)wt[i];

avgtr = avgtr + (float)turn[i];

}

avgwt = avgwt/n;

avgtr = avgtr/n;

RESULTS :

The Result belongs to the algorithm you choose among two of Real time Scheduling algorithms. For both of them the output shows the scheduling of processes one by one and also shows the starting time, ending time, turnaround time and waiting time along with the burst time and dead time given by the user. Also the output is stored in a text file created by the algorithm itself.   
  
  
REFERENCES:

Book (Operating System Concepts Edition 7)

<http://www.cs.ru.nl/lab/rtai/exercises/9.RMS-EDF/Exercise-9.html>

<http://www.cs.bu.edu/techreports/pdf/1998-013-srms-linux-implementation.pdf>