FINAL PROJECT OF OPERATING SYSTEM

1 "An Improved Round Robin Scheduling Algorithm for CPU scheduling"

Abstract:
As we know that operating systems provide much functionality like process management I/O management etc. In the same way, process management is important because when the program run in the operating system then processes access to the hardware so different CPU scheduling has different limitations and weaknesses, therefore, to make efficient the utilization and speed fast of CPU we try to improve Round Robin.

Introduction:
CPU is an important resource of Computers therefore if we want to use maximum utilization of CPU then we have to keep multiple processes in the main memory we have to use the concept of multiprogramming and multitasking and also use the CPU scheduling algorithms.
round-robin is one of the CPU scheduling algorithms which are doing swapping between two queues one is the ready queue and the other is the running queue and context switching takes place amongst the various process and context switching on the bases of time quantum (Fixed time), And if the burst time of the process is finished then it contexts switching to another process and the CPU is not kept idle. The problem with this algorithm is that the process that has very less burst time has to wait too much in the ready queue due to this problem the waiting time and turn round time are increased.
Improved round robin is that there are also two queues one ready queue and a running queue. In the first time, all the processes get the CPU as same as in the round-robin but in the next cycle, the CPU gets that process that has less burst time after this cycle context switching happens CPU gets those whose second number less burst time this procedure going on until all the process finished. In this, the waiting time and turn round time decreased.

Literature:
CPU is an important resource for computers so we have to improve CPU utilization and speed using the concept of multiprogramming
According to M.P Chen in the last 30 years, many resources take place on the topic of the disk-scheduling algorithm.
According to Galvin multiprogramming is a fundamental role in the operating system because the reason behind it is not leaving idle to the CPU and it is doing context switching between the different processes.
According to the Sabrina, the operating system schedules every computer resource before use so it is scheduled also as CPU therefore CPU scheduling is used how to select the process amongst the multiple processes and it is important because it is effect CPU utilization and speed

As we know, we have different algorithms of CPU scheduling from which properties we can check which algorithm is better and these properties are given below.

1. CPU utilization
2. Turnaround time
3. Waiting time
4. Response time
5. Throughput

1. **Simple round-robin:**

We can understand the contemporary RR scheduling algorithm by giving the below steps:
1. The scheduler maintains a queue of ready processes and a list of blocked and swapped-out processes.
2. The PCB of the newly created process is added to the end of the ready queue. The PCB of the terminating process is removed from the scheduling data structures.
3. The scheduler always selects the PCB at the head of the ready queue.
4. When a running process finishes its slice, it is moved to the end of ready queue.
5. the event handler perform the following action
a) When a process makes an I/O request or swapped out, its PCB is removed from the ready queue to blocked/swapped out list.
b) When I/O operation awaited by a process finishes or process is swapped in its PCB is removed from blocked/swapped list to the end of the ready queue.

1. **Proposed algorithm**

1. Allocate all processes to the CPU only one time as like present Round Robin scheduling algorithm.
2. After first time we select shortest job from the waiting queue and it shortest job assign to the CPU.
3. After that we select next shortest job and do step 2
4. Till the complete execution of all processes we repeat steps 2 and 3 that means while all the processes has not been finished(executed).

Experiment:
Here we are taking one experiment and solving the problem with the simple round-robin and with the help of an improved round-robin.

1. **Simple round-robin**

PROCESS	BURST TIME	COMPLETE-TIME	TURN OVER TIME	WAITING TIME
A	24	69	69	45
B	20	65	65	45
C	8	40	40	32
D	10	45	45	35
E	3	23	23	20
F	5	27	27	22

The quantum time=2

A	B	C	D	E	F	A	B	C	D	A	B	A	B	A

Gantt chart=

Average of TURN AROUND TIME=269/6 =44.83333
Average of WAITING TIME=199/6 =33.1616

1. **Improved Round Robin**

PROCESS	BURST TIME	COMPLETE-TIME	TURN OVER TIME	WAITING TIME
A	24	69	69	45
B	20	50	50	35
C	8	30	30	22
D	10	35	35	25
E	3	23	23	20
F	5	27	27	22

The quantum time=2

A	B	C	D	E	F	C	D	B	B	B	A	A	A	A

Gantt chart=

Average of TURN AROUND TIME=234/6 =39
Average of WAITING TIME=164/6 =27.1616

These results show that an improved round robin is better than simple round robin.

Conclusion:

As we know that improved round robin is better than simple round robin but simple round robin is make for time sharing system but that improved round robin is not for time sharing system so in the future we will try to make it of that system.