4.Round Robin Scheduling Algorithm

The Round Robin algorithm is designed especially for time sharing system. It is similar to FCFS but preemption is added switch between processes. A small unit of time called time quantum (or time slice) is defined. A time quantum is generally from 10 to 100 milliseconds. The ready queue is treated as a circular queue, allocated the CPU to each process for a time interval of up to 1 time quantum.
The CPU scheduler picks the first process from the ready queue, sets a timer to interrupt after 1 time quantum, and dispatch the processes. One of two things will then happen. The processes may have a CPU burst of less than 1 time quantum. In this case, the processes itself will release the CPU. The scheduler will then proceed to the next process in the ready queue. Otherwise, if the CPU burst of the currently running processes is longer than 1 time quantum, the timer will go off and will cause an interrupt to the operating system. A context switch will be executed, and the process will be put at the tail of the ready queue. The CPU scheduler will then select the next process in the ready queue.
As an example consider the following set of processes with the length of the CPU burst given in millisecond:

Example 8 using RR: quantum=4

Process	Burst time
P1	24
P2	3
P3	3

Gantt chart:

0 4 7 10 14 18 22 26 30

P1	**P2**	**P3**	**P1**	**P1**	**P1**	**P1**	**P1**

W. T p1 = 26-4-4-4-4-4 = 6
W. T p2 = 4-0 = 4
W. T p3 = 7-0 = 7
Average wait time = (6 + 4 +7) / 3 = 5.66 milliseconds

Example 9 using **preemptive RR**: quantum=2

Process	Burst time	Arrival time
P0	6	0
P1	8	1
P2	4	7
P3	2	9
P4	10	11

0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30

P0	**P1**	**P0**	**P1**	**P2**	**P3**	**P4**	**P0**	**P1**	**P2**	**P4**	**P1**	**P4**	**P4**	**P4**

W. T p0 = 14- 2 – 2- 0= 10
W. T p1 = 22 -2-2-2-1 = 15
W. T p2 = 18 -2-7 = 9
W. T p3 = 10 -9 = 1
W. T p4 = 28 -2-2-2-2-11 = 9

Average wait time = (10 + 15 +9+1+9) / 5 = 8.8 milliseconds

Comparison among Scheduling Algorithms

Algorithms	**Policy type**	**Dis advantages**	**Advantages**
FCFS	Non preemptive	Average waiting time is often quite long.	Easy to implement.
SJF	Non preemptive Or preemptive	Knowing the length of the next CPU request.	Gives minimum average waiting time.
Priority	Non preemptive Or preemptive	Blocking or starvation.	1-Simplicity. 2-support for priority.
R.R	preemptive	If the set time is too long, then the system may become unresponsive, time wasting and would emulate First Come First Served.	It is easy to implement in software