1).FCFS

FCFS(process\_list)

1. initialize(retire\_queue)

2. while(!process\_list.empty())

3. $process = process\_list.pop();

4. $time1=timer.start();

5. send $process to processor for execution;

6. $time2=timer.stop();

7. $process.executionTime=$time2-$time1;

8. send $process to retire\_queue;

9. return retire\_queue;

2).SRT

SRT(process\_list)

1. initialize(ready\_queue, retire\_queue);

2. while(!process\_list.empty() ||! ready\_queue.empty())

3. if(!process\_list.empty())

4. $process = process\_list.pop();

5. put $process in ready\_queue;

6. ready\_queue.ascending\_sort(runtime);

7. $process=ready\_queue.pop();

8. send $process to processor for execution till next process comes;

9. decrement that much amount of time from $process.runtime;

10. if($process.runtime!=0)

11. put $process in ready\_queue after execution ;

12. else

13. $process.executionTime = currentTime - $process.admitTime;

14. send $process in retire\_queue;

15. return retire\_queue;

3).RR

RR(process\_list, $quantum)

1. initialize(ready\_queue, retire\_queue);

2. $pointer = 0;

3. while(!process\_list.empty() ||! ready\_queue.empty())

4. if(!process\_list.empty())

5. $process = process\_list.pop();

6. put $process in ready\_queue;

7. if($pointer==ready\_queue.size())

8. $pointer = 0;

9. if(!ready\_queue.empty())

10. $process=ready\_queue[$pointer];

11. send $process to processor for execution till $quantum is passed;

12. $process.runtime = $process.runtime - 1;

13. if($process.runtime!=0)

14. put $process in ready\_queue after execution ;

15. else

16. $process.executionTime = currentTime - $process.admitTime;

17. send $process in retire\_queue;

18. $pointer++;

19. return retire\_queue;

4).PS

PS(process\_list)

1. initialize(ready\_queue, retire\_queue);

2. while(!process\_list.empty() ||! ready\_queue.empty())

3. if(!process\_list.empty())

4. $process = process\_list.pop();

5. put $process in ready\_queue;

6. ready\_queue.descending\_sort(priority)

7. $process=ready\_queue.pop();

8. send $process to processor for execution till next process comes;

9. decrement that much amount of time from $process.runtime;

10. if($process.runtime!=0)

11. put $process in ready\_queue after execution;

12. else

13. $process.executionTime = currentTime - $process.admitTime;

14. send $process in retire\_queue;

15. return retire\_queue;

6).EDF

EDF(external\_interrupt)

1. initialize(ready\_queue, retire\_queue);

2. while(external\_interrupt ||! ready\_queue.empty())

3. take $process from external interrupt only if its deadline can be fulfilled;

4. put $process in ready\_queue;

5. ready\_queue.ascending\_sort(finish\_deadline);

6. $process=ready\_queue.pop();

7. send $process to processor for execution;

8. wait(till $process completes execution);

9. $process.executionTime = currentTime - $process.admitTime;

10. send $process in retire\_queue;

11. return retire\_queue;

6).RMS

RMS(process\_list)

1. initialize(ready\_queue, retire\_queue);

2. while(external\_interrupt ||! ready\_queue.empty())

3. take $process from external interrupt only if its deadline can be fulfilled;

4. put $process in ready\_queue;

5. ready\_queue.ascending\_sort(process\_period);

6. $process=ready\_queue.pop();

7. send $process to processor for execution;

8. wait(till $process completes execution);

9. $process.executionTime = currentTime - $process.admitTime;

10. send $process in retire\_queue;

11. return retire\_queue;