

# Pseudocode:

## Feedback Algorithms:

**Run():**

While the process queue is not empty, or the ready multi-queue is not empty, or the current process is still running.

Call UpdateQueue().

If the current process is not null:

If the current process is still running:

If the time quantum has reached an interrupt point:

GetNextProcess().

Allow the current process to execute for a time unit.

If not:

Allow the current process to execute for a time unit.

If not:

Reset the time quantum.

GetNextProcess().

If not:

GetNextProcess().

Reset the simulation timer.

Indicate the algorithm has completed.

**UpdateQueues():**

If the process queue is not empty:

If the arrival time of the next process in the process queue is less than or equal to the simulation timer value:

Poll the next process in the process queue and add it to the ready multi-queue.

**GetNextProcess():**

If the ready multi-queue is not empty:

If the current process is not null and the current process is not finished:

Increment the number of times the current process has been preempted.

Place the current process back in the ready multi-queue.

Poll the next process in the ready multi-queue and assign it to the current process.

If not:

If the current process is null or the current process is finished:

Move the timer forward one unit.

## FCFS Algorithm:

**Run():**

While the process queue is not empty, or the ready queue is not empty, or the current process is still running.

Call UpdateQueue().

If the current process is not null:

If the current process is still running:

Allow the current process to execute for a time unit.

If not:

GetNextProcess().

If not:

GetNextProcess().

Reset the simulation timer.

Indicate the algorithm has completed.

**UpdateQueues():**

If the process queue is not empty:

If the arrival time of the next process in the process queue is less than or equal to the simulation timer value:

Poll the next process in the process queue and add it to the ready queue.

**GetNextProcess():**

If the ready queue is not empty:

Poll the next process in the ready queue and assign it to the current process.

If not:

Move the timer forward one unit.

## HRRN Algorithm:

**Run():**

While the process queue is not empty, or the ready queue is not empty, or the current process is still running.

Call UpdateQueue().

If the current process is not null:

If the current process is still running:

If the time quantum has reached an interrupt point:

GetNextProcess().

Allow the current process to execute for a time unit.

If not:

Allow the current process to execute for a time unit.

If not:

Reset the time quantum.

GetNextProcess().

If not:

GetNextProcess().

Reset the simulation timer.

Indicate the algorithm has completed.

**UpdateQueues():**

If the process queue is not empty:

If the arrival time of the next process in the process queue is less than or equal to the simulation timer value:

Poll the next process in the process queue and add it to the ready queue.

**GetNextProcess():**

If the ready queue is not empty:

Poll the next process in the ready queue and assign it to the current process.

If not:

Move the timer forward one unit.

## Round Robin Algorithms:

**Run():**

While the process queue is not empty, or the ready queue is not empty, or the current process is still running.

Call UpdateQueue().

If the current process is not null:

If the current process is still running:

Allow the current process to execute for a time unit.

If not:

GetNextProcess().

If not:

GetNextProcess().

Reset the simulation timer.

Indicate the algorithm has completed.

**UpdateQueues():**

If the process queue is not empty:

If the arrival time of the next process in the process queue is less than or equal to the simulation timer value:

Poll the next process in the process queue and add it to the ready queue.

**GetNextProcess():**

If the ready queue is not empty:

If the current process is not null and the current process is not finished:

Increment the number of times the current process has been preempted.

Place the current process back in the ready queue.

Poll the next process in the ready queue and assign it to the current process.

If not:

If the current process is null or the current process is finished:

Move the timer forward one unit.

## SPN Algorithm:

**Run():**

While the process queue is not empty, or the ready queue is not empty, or the current process is still running.

Call UpdateQueue().

If the current process is not null:

If the current process is still running:

Allow the current process to execute for a time unit.

If not:

GetNextProcess().

If not:

GetNextProcess().

Reset the simulation timer.

Indicate the algorithm has completed.

**UpdateQueues():**

If the process queue is not empty:

If the arrival time of the next process in the process queue is less than or equal to the simulation timer value:

Poll the next process in the process queue and add it to the ready queue.

**GetNextProcess():**

If the ready queue is not empty:

Poll the next process in the ready queue and assign it to the current process.

If not:

Move the timer forward one unit.

## SRT Algorithm:

**Run():**

While the process queue is not empty, or the ready queue is not empty, or the current process is still running.

Call UpdateQueue().

If the current process is not null:

If the current process is still running:

Allow the current process to execute for a time unit.

If not:

GetNextProcess().

If not:

GetNextProcess().

Reset the simulation timer.

Indicate the algorithm has completed.

**UpdateQueues():**

If the process queue is not empty:

If the arrival time of the next process in the process queue is less than or equal to the simulation timer value:

If the current process is not null:

If the service time of the next process in the process queue is less than the remaining service time for the current process:

If the current process is not finished

Increment the number of times the current process has been preempted.

Place the current process back in the ready queue.

Poll the next process in the process queue and add it to the ready queue.

If not:

Poll the next process in the process queue and place it in the ready queue.

If not:

Poll the next process in the process queue and place it in the ready queue.

**GetNextProcess():**

If the ready queue is not empty:

Poll the next process in the ready queue and assign it to the current process.

If not:

Move the timer forward one unit.