Volume 4, Issue 6, June 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Augmented SJF algorithm with reduced starvation

Himani Aggarwal
M-Tech CSE
Geeta Engineering College, Naulatha, India

Er. Shakti Nagpal
HOD, CSE Department
Geeta Engineering College, Panipat, India

Abstract: The main aim of this paper is to develop the new approach for SJF scheduling algorithm which help to reduce the problem of starvation in a heavily loaded computer system. The ASJF algorithm reduce the starvation problem of the simple SJF architecture. This ASJF algorithm is based on SJF and multilevel feedback queue scheduling (MFQS) technique. Comparative analysis of SJF and ASJF is also provided to illustrate the performance differences in them.

Keywords: CPU Scheduling, SJF, Scheduling Algorithm, Turnaround time, Waiting Time, Response Time, Starvation, Gantt Chart

I. Introduction

Scheduling is the method by which threads, processes or data flows are given access to system resources. The need for a scheduling algorithm arises from the requirement for most modern systems to perform multitasking (execute more than one process at a time) and multiplexing (transmit multiple flows simultaneously). Scheduling is a fundamental operating-system function. Almost all computer resources are scheduled before use. The CPU is, of course, one of the primary computer resources. Thus, its scheduling is central to operating-system design.

II. Scheduling Criteria:

a) CPU Utilization: (max) To keep the CPU as busy as possible.
b) Turnaround time: (min) Amount of time to execute a particular process. The interval from the time of submission of a process to the time of completion is the turnaround time
c)Response time: (min) Time taken from first request until first response produced. It is the time it takes to start responding, not the time it takes to output the response.
d)Waiting time: (min)Amount of time a process waiting in the ready queue. The CPU scheduling algorithm affects only the amount of time that a process spends waiting in the ready queue.
e)Throughput: (max) No. of processes that complete their execution per unit of time.
f)Starvation: (min) Blocking of processes with the high service time requests. Process should not experience an unbounded wait time before or while process service. It is also known as Indefinite Blocking.
According to Silberchatz, Galvin and Gagne, A major problem with SJF scheduling algorithms is indefinite blocking, or starvation. It may penalize the processes with high service time requests. If the ready list is saturated , then processes with the large service times tend to be left in the ready list while small processes receive service. In extreme case where the system has little idle time processes with the large service time will never be served. This total starvation of large processes may be a serious liability of this algorithm. A solution to this problem is aging. Aging is a technique of gradually increasing the priority of processes that wait in the system for a long time.

Scheduling objectives:
a)Avoid indefinite postponement & starvation in SJF scheduling algorithm.
b)Optimize the system performance and maximize CPU utilization and throughput.
c)Retain fairness in the resource allocation.
d)Minimize turnaround time, response time, starvation and waiting time.
e)Effectively deal with the current trend of more instead of faster processors.

Simple Shortest Job First (SJF) Algorithm
According to Silberchatz, Galvin, Gagne in operating system design and operating system by D M Dhamdhere, the simple SJF scheduling algorithm is given by following steps:-
1. The scheduler maintains a queue of ready processes and a list of blocked and swapped out processes.
2. The PCB of newly created process is added to end of ready queue. The PCB of terminating process is removed from the scheduling data structures.

3. The scheduler always selects the PCB at head of the ready queue.
4. When a running process finishes its slice, it is moved to end of ready queue.
5. The event handler perform the following action

a) When a process makes an input -output request or swapped out, its PCB is removed from ready queue to blocked/swapped out list

b) When input-output operation awaited by a process finishes or process is swapped in its process control block is removed from blocked/swapped list to end of ready queue.

Augmented SJF (ASJF) scheduling algorithm

The ASJF algorithm will reduce the problem of starvation as well as helps to attack both efficiency and response time problems. MFQS will help in giving a new run able process a higher priority and shorter time slice. Priority of queue & processes, process no. & burst time of processes are predefined. The algorithm will be performed as these steps:

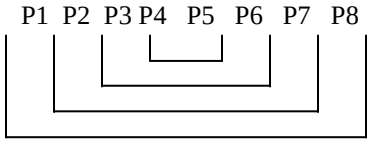
a) Separate processes according to their CPU bursts. If a process uses too much CPU time, it will be moved to a lower-priority queue. This scheme leaves I/O-bound and interactive processes in the higher-priority queues.

b)A process that waits too long in a lower-priority queue may be moved to a higher-priority queue. A process that takes too long to execute must be moved to lower priority queue.

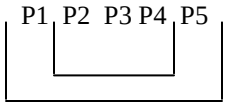
c)The tasks are sorted according to their next CPU bursts. The process with the lowest CPU burst or execution time is allocated first and those with the same execution time are handled on the basis of First Come First Serve technique.

d)Allow all the processes to move between queues.Allocate all sorted process to CPU for a predetermined time slice. Value of time slice= (Burst time of shortest process+Burst time of largest process)/2

e) After all the processes have been executed once using the given time slice, swap the process priorities symmetrically with other process as:



(when no of processes=8)



(when no of processes=5, priority of P3 remains same)

III. Comparative study

For illustrating ASJF proposed algorithm, five processes are taken and are scheduled according to SJF and ASJF both. The response time, waiting time for longer process among all, context switch, and throughput is calculated and the results were compared.

Experiment 1: Here, we consider the processes each with its priority and service time.

ProcessID	Service time	Priority
P1	86	7
P2	54	5
P3	49	4
P4	34	9
P5	92	2

In SJF

P5	P3	P2	P1	P4
0 92	141	195	281	315

In ASJF,
Run each process with the
Time slice=(92+34)/2=126/2=63 ms

P5	P3	P2	P1	P4

0 63 112 166 229 263

P3, P2 and P4 are executed completely. Time left for the execution of process:
P5=92-63=29 ms and P1=86-63=23 ms
In ASJF uncompleted processes are synchronized. Thus, P5, P1 becomes P1, P5.

P5	P3	P2	P1	P4	P1	P5

0 63 112 166 229 263 286 315

ProcessID	Response Time in SJF	Response Time in ASJF(ms)
P5	0	0
P3	92	63
P2	141	112
P1	195	166
P4	281	229

Waiting time for longest process P1is 195 in SJF and 166 in ASJF
Context switch in SJF is 4 and ASJF is 7.
Throughput in SJF and ASJF is same
= 5 / 315 = 0.0158730

Experiment 2: Here, we consider the processes each with its priority and service time.

ProcessID	Service time	Priority
P1	10	1
P2	40	4
P3	20	3
P4	30	2
P5	60	5
P6	50	6

In SJF,

P1	P4	P3	P2	P5	P6

0 10 40 60 100 160 210

In ASJF,
Run each process with the
Time slice=(60+10)/2=70/2=35 ms

P1	P4	P3	P2	P5	P6

0 10 40 60 95 130 165

P1 P4and P3are executed completely. Time left for the execution of process:
P2=40-35=5 ms, P5=60-35=25 ms and P6=50-35=15 ms

In ASJF uncompleted processes are synchronized. Thus, P2, P5, P6 becomes P6, P5, P2.

P1	P4	P3	P2	P5	P6	P6	P5	P2

0 10 40 60 95 130 165 180 205 210

ProcessID	Response Time in SJF	Response Time in ASJF(ms)
P1	0	0
P4	10	10
P3	40	40
P2	60	60
P5	100	95
P6	160	130

Waiting time for longest process P5is 100 in SJF and 95 in ASJF.
Context switch in SJF is 5 and ASJF is 6.
Throughput in SJF and ASJF is same =
$= \frac{6}{210} = 0.02857$

Experiment 3: Here, we consider the processes each with its priority and service time.

ProcessID	Service time	Priority
P2	22	1
P1	35	2
P3	54	3
P4	68	4
P6	72	5
P5	79	6

In SJF

P2	P1	P3	P4	P6	P5

0 22 57 111 179 251 330

In ASJF,
Run each process with the
Time slice=(22+79)/2=101/2=50.5ms

P2	P1	P3	P4	P6	P5

0 22 57 107.5 58 208.5 259

P2 and P1 are executed completely. Time left for the execution of process:
P3=54-50.5=29 ms, P4=68-50.5=17.5, P6=72-50.5=21.5 and P5=79-50.5=28.5

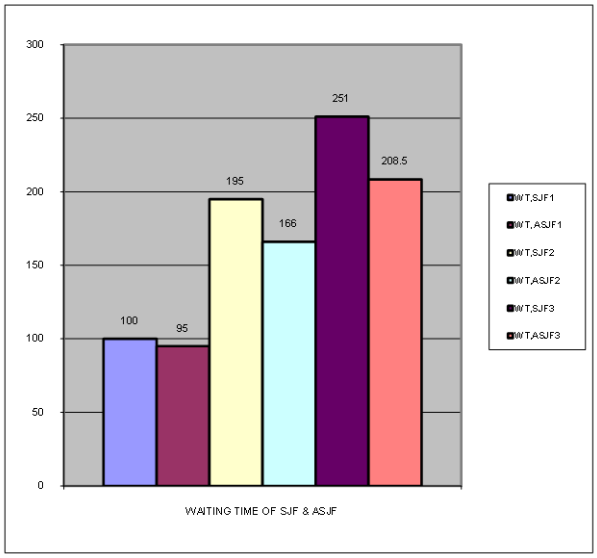
In ASJF uncompleted processes are synchronized. Thus, P3, P4,P6,P5 becomes P5,P6,P4,P3.

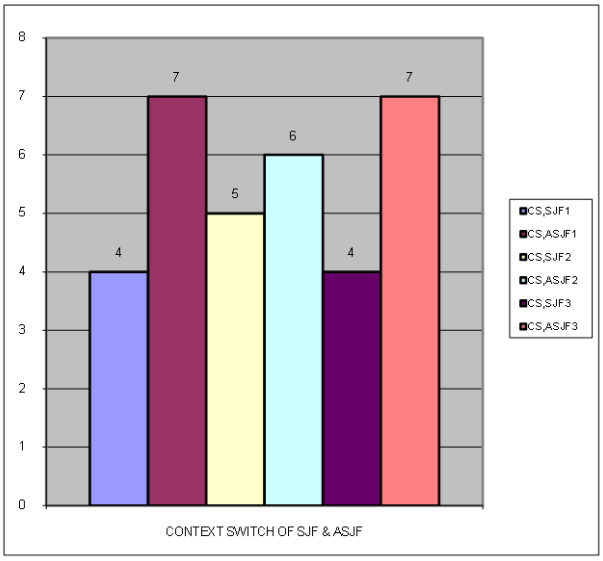
P2	P1	P3	P4	P6	P5	P5	P6	P4	P3

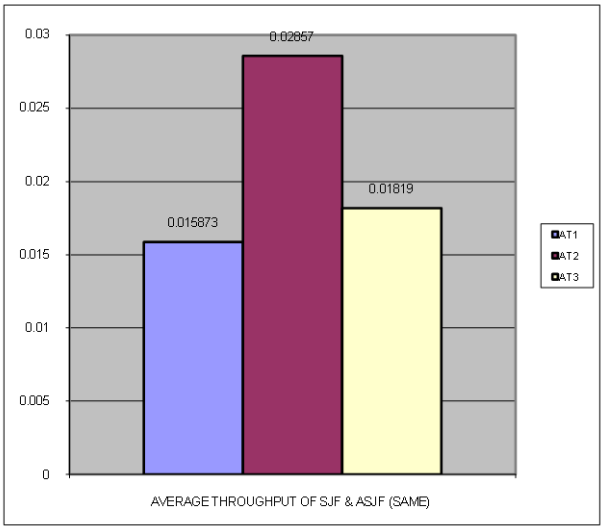
0 22 57 107.5 158 208.5 259 287 309 326.5 330

ProcessID	Response Time in SJF	Response Time in ASJF(ms)
P2	0	0
P1	22	22
P3	57	57
P4	111	107.5
P6	179	58.0
P5	251	208.5

Waiting time for longest process P5is 251 in SJF and 208.5 in ASJF
Context switch in SJF is 4 and ASJF is 7.
Throughput in SJF and ASJF is same =
NO. of jobs = 6 = 0.01819Total Time







IV. Conclusion and Future scope:

We made a comparative study of SJF and ASJF algorithm. It is concluded that the ASJF algorithm is superior in terms of minimizing degree of starvation, increasing fairness, decrease in response time and timely resource allocation to individual process**.** ASJF algorithm clearly shows maximum CPU utilization and efficient handling of resources. As MFQS algorithm is merged with SJF, the technique of dividing the processes to various queues and switching of processes among them will further reduce the problem of starvation. In the future this ASJF scheduling algorithm can be further improved so as to completely remove the problem of starvation from the SJF algorithm.

References:

[1] http://en.wikipedia.org./wiki/scheduling
[2] Operating Systems Sibsankar Haldar 2009,Pearson Education, India
[3] D.M. Dhamdhere operating Systems A Concept Based Approach,Second edition, Tata McGraw-Hill, 2006.
[4] Sabrina, F.C.D, Nguyen, S.jha, D. Platt and F. Safaei,2005.Processing scheduling in programmable networks. Computer commun, 28:676-687.
[5] Silberchatz, Galvin and Gagne, 2003. Operating systems concepts.
[6] S.Sredharran,Operating Systems,2000 ISBN 81-87721-18-9

