

1. CPU调度算法确定其调度进程的执行顺序。给定一个处理器上要调度 n 个进程，那么可能有多少个不同的调度？给出一个关于 n 的公式。

n 的阶乘 ($n!$)

2. 考虑以下一组进程，CPU突发时间的长度以毫秒为单位，回答以下问题：

评分标准：一个算法4分，分到abcd 4个小问题各1分

Process	Burst Time	Priority
P_1	2	2
P_2	1	1
P_3	8	4
P_4	4	2
P_5	5	3

The processes are assumed to have arrived in the order P_1, P_2, P_3, P_4, P_5 , all at time 0.

- Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, non-preemptive priority (a larger priority number implies a higher priority), and RR (quantum = 2).
- What is the turnaround time of each process for each of the scheduling algorithms in part a?
- What is the waiting time of each process for each of these scheduling algorithms?
- Which of the algorithms results in the minimum average waiting time (over all processes)?

a. FCFS, SJF, non-preemptive priority, priority, RR的甘特图

- FCFS

进程	运行时间	优先级	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1	2	2																				
P2	1	1																				
P3	8	4																				
P4	4	2																				
P5	5	3																				

- SJF

进程	运行时间	优先级	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1	2	2																				
P2	1	1																				
P3	8	4																				
P4	4	2																				
P5	5	3																				

- non-preemptive priority

进程	运行时间	优先级	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1	2	2																				
P2	1	1																				
P3	8	4																				
P4	4	2																				
P5	5	3																				

- RR

进程	运行时间	优先级	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1	2	2																				
P2	1	1																				
P3	8	4																				
P4	4	2																				
P5	5	3																				

b. a中process在各个算法下的周转时间

进程	FCFS	SJF	PSA	RR
P1	2	3	15	2
P2	3	1	20	3
P3	11	20	8	20
P4	15	7	19	13
P5	20	12	13	18

c. a中process在各个算法下的等待时间

进程	FCFS	SJF	PSA	RR
P1	0	1	13	0
P2	2	0	19	2
P3	3	12	0	12
P4	11	3	15	9
P5	15	7	8	13

d. a中各个算法的平均等待时间

算法	等待时间
FCFS	6.2
SJF	4.6
PSA	11
RR	7.2

平均等待时间最短的算法是 SJF 调度算法

3. 以下进程使用抢占式RoundRobin调度算法进行调度，回答以下问题：

<u>Process</u>	<u>Priority</u>	<u>Burst</u>	<u>Arrival</u>
P_1	40	20	0
P_2	30	25	25
P_3	30	25	30
P_4	35	15	60
P_5	5	10	100
P_6	10	10	105

1. 使用甘特图显示进程的调度顺序

进程	到达时间	运行时间	优先级	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	105	110	115	120
P1	0	20	40																								
P2	25	25	30																								
P3	30	25	30																								
P4	60	15	35																								
P5	100	10	5																								
P6	105	10	10																								
Pidle	0	inf	0																								

2. 每个进程的周转时间是多少

进程	周转时间
P1	20
P2	55
P3	60
P4	15
P5	20
P6	10

3. 每个进程的等待时间是多少

进程	等待时间
P1	0
P2	30
P3	35
P4	0
P5	10
P6	0

4. CPU利用率是多少

- $\frac{105}{120} \times 100\% = 87.5\%$

4. 以下算法之间的关系

- 优先级算法 和 最短作业优先
 - 优先级算法的优先级设置中，设置作业时间与优先级成反比，作业时间越长，优先级越低，就是最短作业优先算法
- 多级反馈队列 和 先进先出
 - 多级反馈队列中可以使用先进先出作为某一级队列的算法
- 优先级算法 和 先进先出
 - 优先级算法的优先级设置中，设置作业到达时间与优先级成反比，作业到达时间越低，优先级越高，就是先进先出算法
- 时间片轮转 和 最短作业优先

- 时间片设置得越短越接近于最短作业优先，但无法变成最短作业优先

5. I/O密集型和CPU密集型程序，哪个更有可能进行自愿的上下文切换，哪个更有可能进行非自愿的上下文切换吗？解释你的答案。

I/O密集型程序更有可能进行自愿的上下文切换，而CPU密集型程序更有可能进行非自愿的上下文切换。

- 在I/O密集型程序中，由于其主要任务是进行输入/输出操作，会频繁地发起I/O请求并等待I/O完成。当一个I/O请求发出后，程序可以主动选择进行上下文切换，让其他等待执行的进程有机会执行，以充分利用CPU资源。因此，I/O密集型程序更有可能进行自愿的上下文切换。
- CPU密集型程序会长时间占用CPU执行计算密集型任务，通常不会主动放弃CPU。当CPU密集型程序的时间片用尽或者其他高优先级的进程需要执行时，操作系统会强制进行上下文切换，将CPU分配给其他进程。因此，CPU密集型程序更有可能进行非自愿的上下文切换。

6. 以下哪种调度算法可能导致饥饿

b和d

b中，如果不间断有短作业到达，长作业会饥饿

d中，如果不间断有高优先级作业到达，低优先级作业会饥饿

7. 以下进程使用抢占式、基于优先级的时间片轮转调度算法进行调度。

- 使用甘特图显示进程的调度顺序。

进程	到达时间	运行时间	优先级	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95
P1	0	15	8																			
P2	0	20	3																			
P3	20	20	4																			
P4	25	20	4																			
P5	45	5	5																			
P6	55	15	5																			

- 每个进程的周转时间是多少？

进程	周转时间
P1	15
P2	95
P3	55
P4	55
P5	5
P6	15

- 每个进程的等待时间是多少？

进程	等待时间
P1	0
P2	75
P3	35
P4	35
P5	0
P6	0