第三次作业

线程

1. Discuss the difference between user-level thread and kernel level thread.

答：用户级线程由用户空间的库管理，切换速度快且开销小，但内核对这些线程并不知情，可能导致多任务处理时的低效；内核级线程由操作系统管理，能够更好地进行调度，并利用多处理器的优势，这样可以更有效地使用资源，但由于需要内核干预，开销相对较高。

2. Which of the following components of program state are shared across threads in a multithreaded process?

a. Register values

b. Heap memory

c. Global variables

d. Stack memory

答：b、c（堆内存、全局变量）

3. The program shown below uses the Pthreads API. What would be output from the program at LINE C and LINE P?

```
include
#include

int value=0;
void *runner(void *param); /* the thread */

int main()
{
   int pid;
   pthread_t tid;
   pthread_attr_t attr;
   pid=fork();
   if(pid==0)
   {
      pthread_attr_init(&attr);
      pthread_create(&tid, &attr, runner, NULL);
      pthread_join(tid, NULL);
      printf("CHILD: value=%d\n", value);   /* LINE C */
   }else if(pid>0){
      wait(NULL);
      printf("PARENT: value=%d\n",value);   /* LINE P */
   }
}

void *runner(void *param)
{
   value=5;
   pthread_exit(0);
}
```

答：LINE C:CHILD:value=5   LINE P:PARENT:value=0

父进程在创建子进程之前没有修改 value，因此父进程中 value 的值为 0；子进程中，线程修改了全局变量的值为 5，进程中的线程与进程共享全局变量，因此 value 的值为 5。

4. 请说明三种多线程模型及其优缺点。

答：三种多线程模型指多对一模型、一对一模型和多对多模型。

多对一模型是指多个用户级线程映像进单个内核级线程，优点是上下文切换快速，用户级线程管理简单，缺点是并发性低，任一时刻只能有一个线程可以访问内核，且一个线程阻塞会导致整个进程阻塞。

一对一模型是指一个用户级线程映射到一个内核级线程，它提供了更好的并发性，内核能够调度多个线程，充分利用多核处理器，且每个线程都有独立的调度，缺点是每创建一个用户级线程就需要创建一个内核线程，创建和管理线程的开销较大，系统资源消耗高。

多对多模型是指多个用户级线程映射到多个内核级线程，可以根据系统负载动态调整线程数，灵活性高，缺点是实现复杂，调度和管理难度较大。

CPU 调度

1、Consider the following set of processes, with the length of the CPU burst given in milliseconds:

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

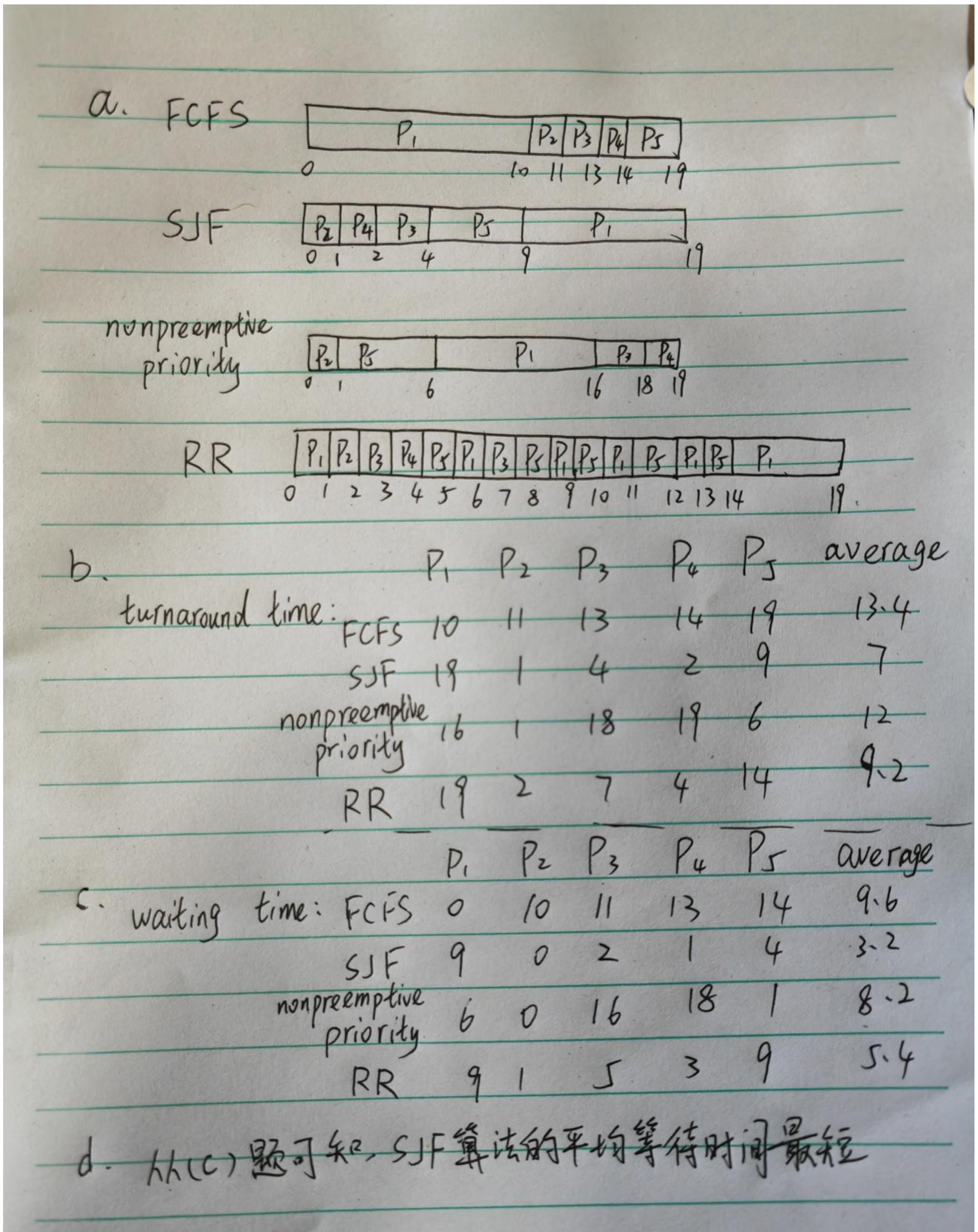
The processes are assumed to have arrived in the order P_1, P_2, P_3, P_4, P_5, all at time 0.

a. Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1).

b. What is the turnaround time of each process for each of the scheduling algorithms in part a?

c. What is the waiting time of each process for each of the scheduling algorithms in part a?

d. Which of the algorithms in part a results in the minimum average waiting time (over all processes)?



a. FCFS

P1	P2	P3	P4	P5
0–10	10–11	11–13	13–14	14–19

SJF

P2	P4	P3	P5	P1
0–1	1–2	2–4	4–9	9–19

nonpreemptive priority

P2	P5	P1	P3	P4
0–1	1–6	6–16	16–18	18–19

RR

P1	P2	P3	P4	P5	P1	P3	P5	P1	P5	P1	P5	P1	P5	P1
0–1	1–2	2–3	3–4	4–5	5–6	6–7	7–8	8–9	9–10	10–11	11–12	12–13	13–14	14–19

b. turnaround time:

	P1	P2	P3	P4	P5	average
FCFS	10	11	13	14	19	13.4
SJF	19	1	4	2	9	7
nonpreemptive priority	16	1	18	19	6	12
RR	19	2	7	4	14	9.2

c. waiting time:

	P1	P2	P3	P4	P5	average
FCFS	0	10	11	13	14	9.6
SJF	9	0	2	1	4	3.2
nonpreemptive priority	6	0	16	18	1	8.2
RR	9	1	5	3	9	5.4

d. 从(c)题可知、SJF算法的平均等待时间最短

2、Which of the following scheduling algorithms could result in starvation?

a. First-come, first-served

b. Shortest job first

c. Round robin

d. Priority

答：b、d（最短作业优先、优先级调度）

最短作业优先算法中，系统总是选择预计运行时间最短的作业，如果有大量短作业不断到达，长作业可能会被长期推迟，导致饥饿；优先级调度中，系统总是选择优先级最高的作业执行，如果高优先级的作业持续到达，低优先级的作业可能会被永远推迟，导致饥饿。