第二章 应用题参考答案

6 若有一组作业 J1，…，Jn，其执行时间依次为 S1，…，Sn。如果这些作业同时到达系统，并在一台单 CPU 处理器上按单道方式执行。试找出一种作业调度算法，使得平均作业周转时间最短。（10 分）

答：首先，对 n 个作业按执行时间从小到大重新进行排序，则对 n 个作业：J1’，…，Jn’，它们的运行时间满足：S1’≤ S2’≤… ≤S(n-1)’≤Sn’。那么有：

$$T=[S_1'+(S_1'+S_2')+(S_1'+S_2'+S_3')+\cdots+(S_1'+S_2'+S_3'+\cdots+S_n')]/n$$
$$=[n\times S_1'+(n-1)\times S_2'+(n-3)\times S_3']+\cdots+S_n']]/n$$
$$=(S_1'+S_2'+S_3'+\cdots+S_n')-[0\times S_1'+1\times S_2'+2\times S_3'+\cdots+(n-1)S_n']/n$$

由于任何调度方式下，$S_1'+S_2'+S_3'+\cdots+S_n'$为一个确定的数，而当 $S_1'\leq S_2'\leq\cdots\leq S_{(n-1)}'\leq S_n'$ 时才有：$0\times S_1'+1\times S_2'+2\times S_3'+\cdots+(n-1)S_n'$的值最大，也就是说，此时 T 值最小。所以，按短作业优先调度算法调度时，使得平均作业周转时间最短。

8 在道数不受限制的多道程序系统中，有作业进入系统后备队列时立即进行作业调度。现有 4 个作业进入系统，有关信息列于下表，当作业调度和进程调度均采用高优先级算法时(规定数大则优先级高)。（10 分）

作业名	进入后备队列时间	执行时间	优先级
JOB1	8:00	60 分	1
JOB2	8:30	50 分	2
JOB3	8:40	30 分	4
JOB4	8:50	10 分	3

试填充下表。

作业名	进入后备队列时间	执行时间	开始执行时间	结束执行时间	周转时间	带权周转时间
平均周转时间 T=						
带权平均周转时间 W=						

解:

作业名	进入后备队列时间	执行时间	开始执行时间	结束执行时间	周转时间	带权周转时间
JOB1	8:00	60 分	8:00	9:00	60	60/60
JOB3	8:40	30 分	9:00	9:30	50	50/30
JOB4	8:50	10 分	9:30	9:40	50	50/10
JOB2	8:30	50 分	9:40	10:30	120	120/50

平均周转时间 T=	(60+50+50+120)/4=70
带权平均周转时间 W=	(1+5/3+5+12/5)/4=2.52

9　对某系统进行监测后表明，每个进程在 I/O 阻塞之前的运行时间为 T。一次进程切换的系统开销时间为 S。若采用时间片长度为 Q 的时间片轮转法，对下列各种情况算出 CPU 利用率。（10 分）

1）Q＝∞　2）Q＞T　3）S＜Q＜T　4) Q＝S　5) Q 接近于 0。

答： 1）Q=∞　CPU 利用率=T/（T+S）

2）Q>T　CPU 利用率=T/（T+S）

3）T>Q>S　CPU 利用率=Q/（Q+S）

4) Q=S　CPU 利用率=50%

5) Q→0　CPU 利用率→0

19　单处理机多道分时系统中，有 3 道作业依次提交：（10 分）

作业	作业提交时间	运行时间	其中	
		(单位：小时)	I/O 时间	CPU 时间
Job1	8.0	0.36	0.18	0.18
Job2	8.2	0.32	0.16	0.16
Job3	8.4	0.36	0.18	0.18

(1)若每道作业的 I/O 等待时间占各自的总运行时间的一半；
(2)分时运行两道作业，CPU 将有 20%的时间空闲；
(3)除 CPU，系统有充足的资源供作业使用。
试计算各作业运行完成时间。

答：分时系统中，作业进程轮流占用 CPU，按时间片轮转。已知单道运行时，I/O 等待时间各占总运行时间的 50%；若两个作业进程分时运行，则 CPU 有 20%的时间空闲、两道运行 CPU 空闲缩短了、即 CPU 为各进程平均运行 40%。结果见下表。

作业 提交时间	提交 并调用	道数	CPU 等待	占用 CPU	运行 时间	作业	进度	还需 时间
8.0-8.2	Job1	1	50%	50%	0.2	Job1	0.1	0.08
8.2-8.4	Job2	2	20%	40%	0.2	Job1 Job2	0.08 0.08	0 0.08
8.4-8.6	Job3	2	20%	40%	0.2	Job2 Job3	0.08 0.08	0 0.1
8.6-8.8		1	50%	50%	0.2	Job3	0.1	0
作业完 成时间	Job1 为 8.4 ， job2 为 8.6 ， job3 为 8.8							

27　某多道程序系统供用户使用的主存为 100K，磁带机 2 台，打印机 1 台。采用可变分区主存管理，采用静态方式分配外围设备，忽略用户作业 I/O 时间。现有作业序列如下：

作业号	进入输入井时间	运行时间	主存需求量	磁带需求	打印机需求
1	8:00	25 分钟	15K	1	1
2	8:20	10 分钟	30K	0	1
3	8:20	20 分钟	60K	1	0
4	8:30	20 分钟	20K	1	0
5	8:35	15 分钟	10K	1	1

作业调度采用 FCFS 策略，优先分配主存低地址区且不准移动已在主存的作业，在主存中的各作业平分 CPU 时间。(共 14 分)

现求：(1)作业被调度的先后次序? (6 分)

(2)全部作业运行结束的时间?　(4 分)

(3)作业平均周转时间为多少?　(2 分)

(4)最大作业周转时间为多少?　(2 分)

答：(1)作业调度选择的作业次序为：作业 1、作业 3、作业 4、作业 2 和作业 5。

(2)全部作业运行结束的时间 9:30。

(3)周转时间：作业 1 为 30 分钟、作业 2 为 55 分钟、作业 3 为 40 分钟、作业 4 为 40 分钟和作业 5 为 55 分钟。

(4)平均作业周转时间=44 分钟。

(5))最大作业周转时间为 55 分钟。

分析：本题综合测试了作业调度、进程调度、及对外设的竞争、主存的竞争。

8:00　作业 1 到达，占有资源并调入主存运行。

8:20　作业 2 和 3 同时到达，但作业 2 因分不到打印机，只能在后备队列等待。作业 3 资源满足，可进主存运行，并与作业 1 平分 CPU 时间。

8:30　作业 1 在 8:30 结束，释放磁带与打印机。但作业 2 仍不能执行，因不能移动而没有 30KB 的空闲区，继续等待。作业 4 在 8:30 到达，并进入主存执行，与作业 3 分享 CPU。

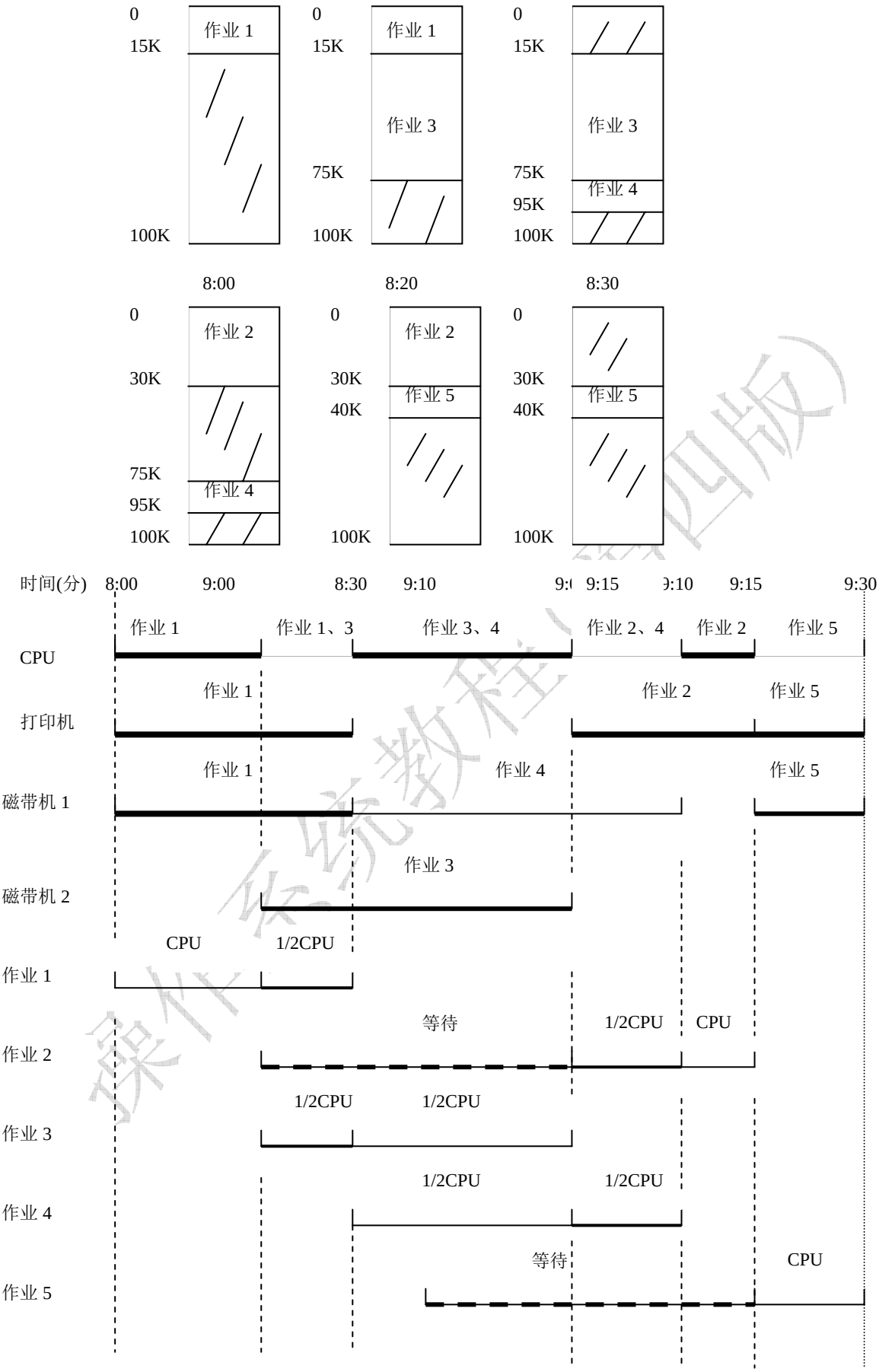
8:35　作业 5 到达，因分不到磁带机/打印机，只能在后备队列等待。

9:00　作业 3 运行结束，释放磁带机。此时作业 2 的主存及打印机均可满足，投入运行。作业 5 到达时间晚，只能等待。

9:10　作业 4 运行结束，作业 5 因分不到打印机，只能在后备队列继续等待。

9:15　作业 2 运行结束，作业 5 投入运行。

9:30　作业全部执行结束。


0
15K
100K
作业 1
作业 3
75K
作业 4
95K
8:00
8:20
8:30
0
30K
40K
75K
95K
100K
作业 2
作业 5
时间(分)
8:00
9:00
8:30
9:10
9:15
9:30
CPU
作业 1
作业 1、3
作业 3、4
作业 2、4
作业 2
作业 5
打印机
磁带机 1
磁带机 2
作业 4
CPU
1/2CPU
等待
1/2CPU
CPU
作业 1
作业 2
作业 3
作业 4
作业 5

29　上题中，若允许移动已在主存中的作业，其他条件不变，现求：(共 16 分)

(1)FIFO 算法选中作业执行的次序及作业平均周转时间。(8 分)

(2) SJF 算法选中作业执行的次序及作业平均周转时间。(8 分)

答：

解：1.先来先服务算法。说明：

(1) 8:30　作业 A 到达并投入运行。注意它所占用的资源。

(2) 8:50　作业 B 到达，资源满足进主存就绪队列等 CPU。

(3) 9:00　作业 C 到达，主存和磁带机均不够，进后备作业队列等待。

(4) 9:05　作业 D 到达，磁带机不够，进后备作业队列等待。后备作业队列有 C、D。

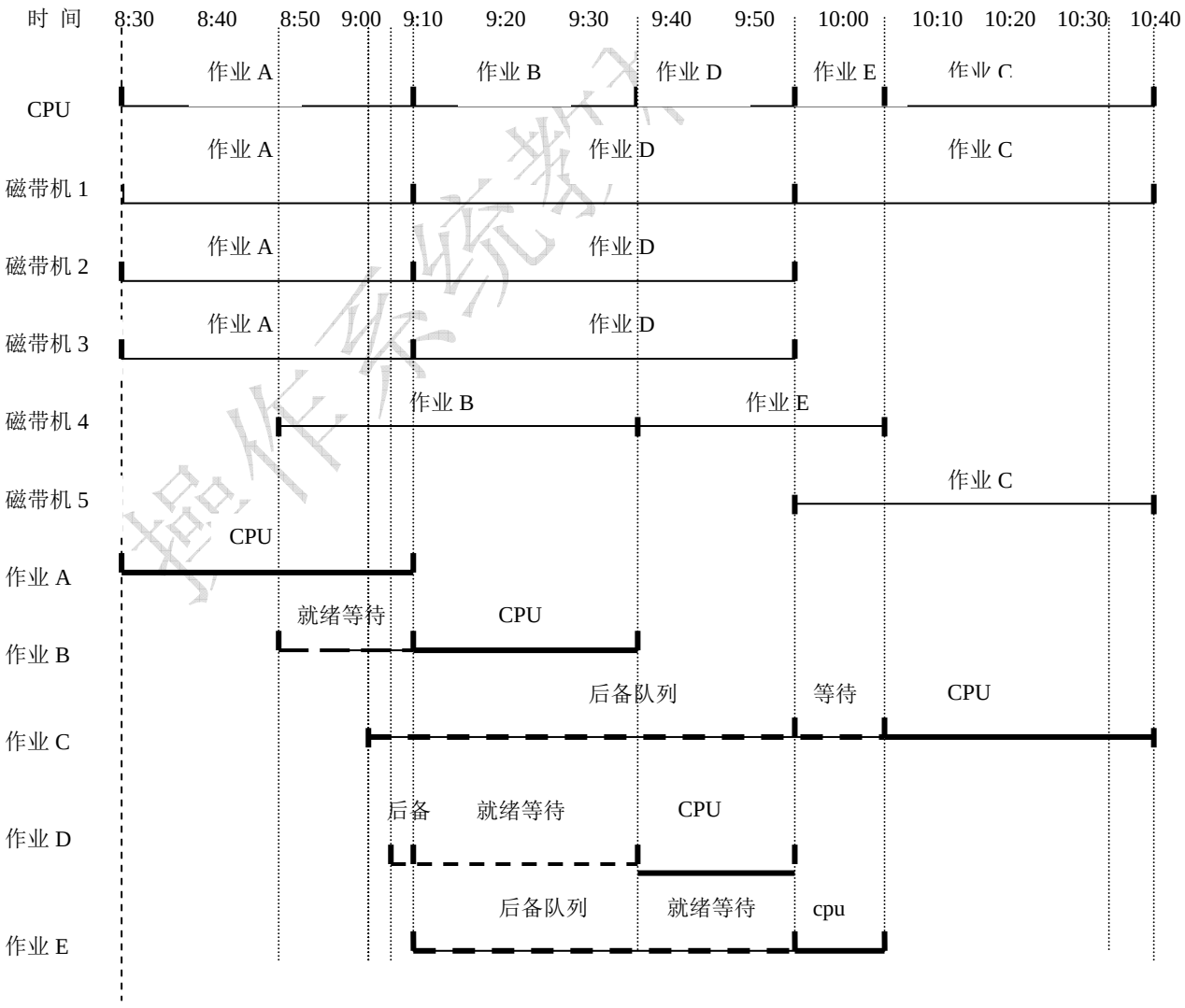
(5) 9:10　作业 A 运行结束，归还资源磁带，注意归还的主存能移动，这样主存中空出了 80KB 的空闲区。作业 B 投入运行。这时作业 E 也到达了，后备作业队列中依次有作业 C、D、E。这时作业 D 和作业 E 都满足条件，均进主存就绪队列等 CPU。

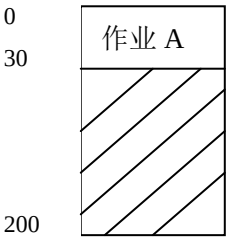
(6)9:35　作业 B 运行结束，作业 D 投入运行。这时作业 C 因磁带不满足而继续在后备作业队列等待。

(7)9:55　作业 D 运行结束，作业 E 投入运行。这时作业 C 因资源满足而调入主存进就绪队列等 CPU。

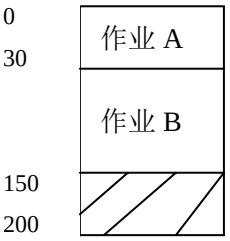
(8)10:05　作业 E 运行结束，作业 C 投入运行。

(9)10:40　作业 C 运行结束。

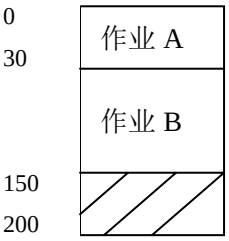




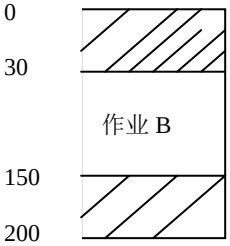
8:30



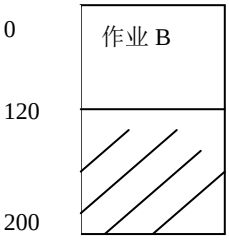
8:50



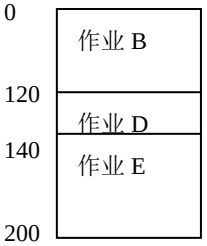
9:00,9:05



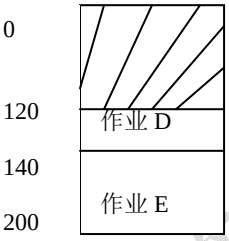
9:10



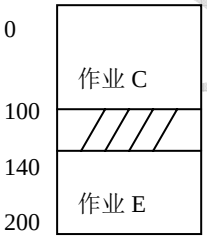
9:10



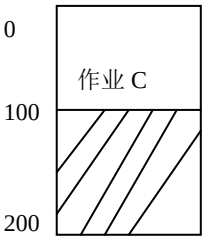
9:10



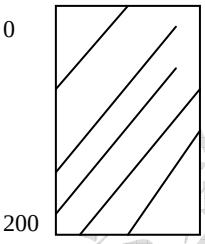
9:35



9:55



10:05



10:40

作业执行次序	进输入井时间	装入主存时间	开始执行时间	执行结束时间	周转时间
作业 A	8:30	8:30	8:30	9:10	40(分)
作业 B	8:50	8:50	9:10	9:35	45
作业 D	9:05	9:10	9:35	9:55	50
作业 E	9:10	9:10	9:55	10:05	55
作业 C	9:00	9:55	10:05	10:40	100
作业平均周转时间		(40+45+50+55+100)/5=58 分钟			

作业执行次序为 A、B、D、E、C。

2. 短作业优先算法。说明：

(1) 8:30　　作业 A 到达并投入运行。注意它所占用的资源。

(2) 8:50　　作业 B 到达，资源满足进主存就绪队列等 CPU。

(3) 9:00　　作业 C 到达，主存和磁带机均不够，进后备作业队列等待。

(4) 9:05　　作业 D 到达，磁带机不够，进后备作业队列等待。后备作业队列有 C、D。

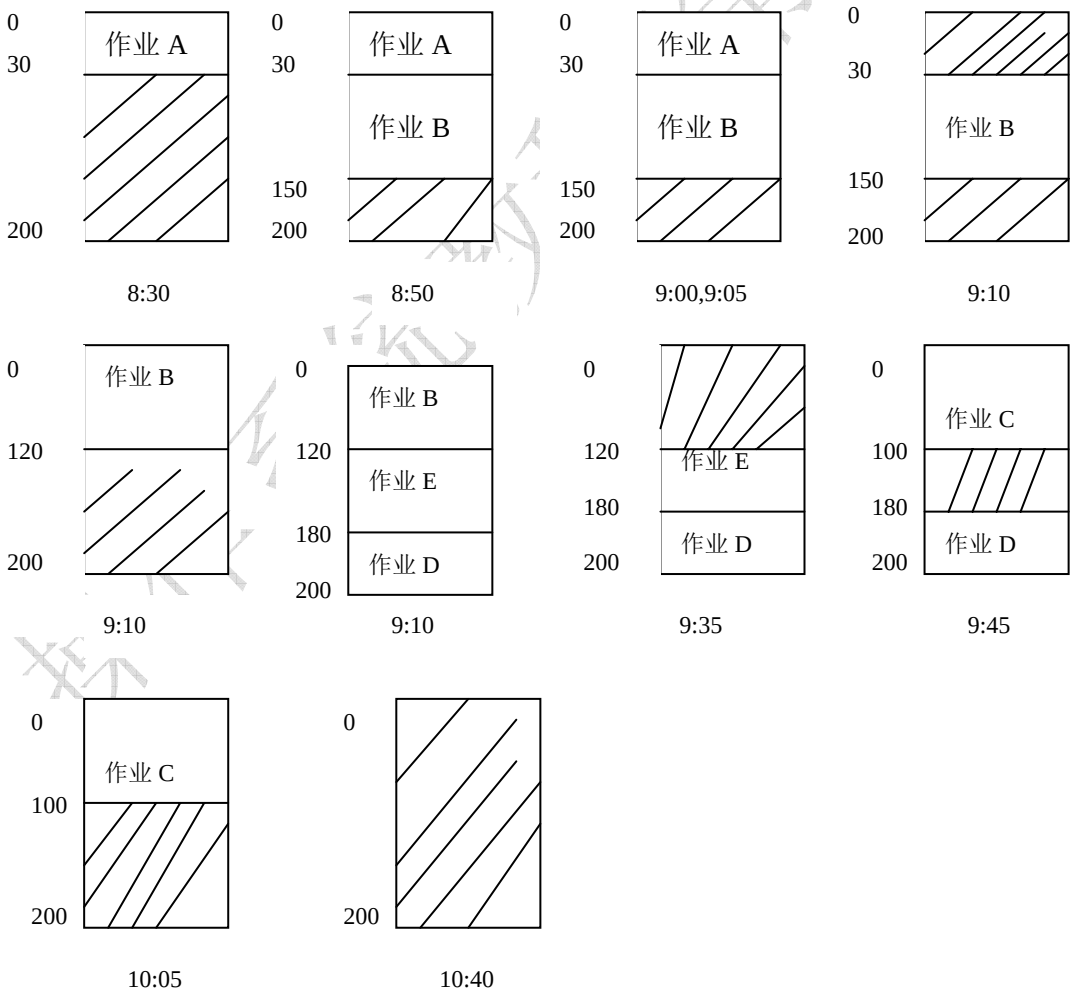
(5) 9:10　　作业 A 运行结束，归还资源磁带，但注意主存能移动。作业 B 投入运行。由于作业 E 也到达后备队列，后备作业队列有作业 C、D、E 等待。这时已有 80KB 主存可用，按 SJF，先调作业 E，再调作业 D 进主存就绪队列等待。而作业 C 因主存和磁带均不足继续等在后备队列。

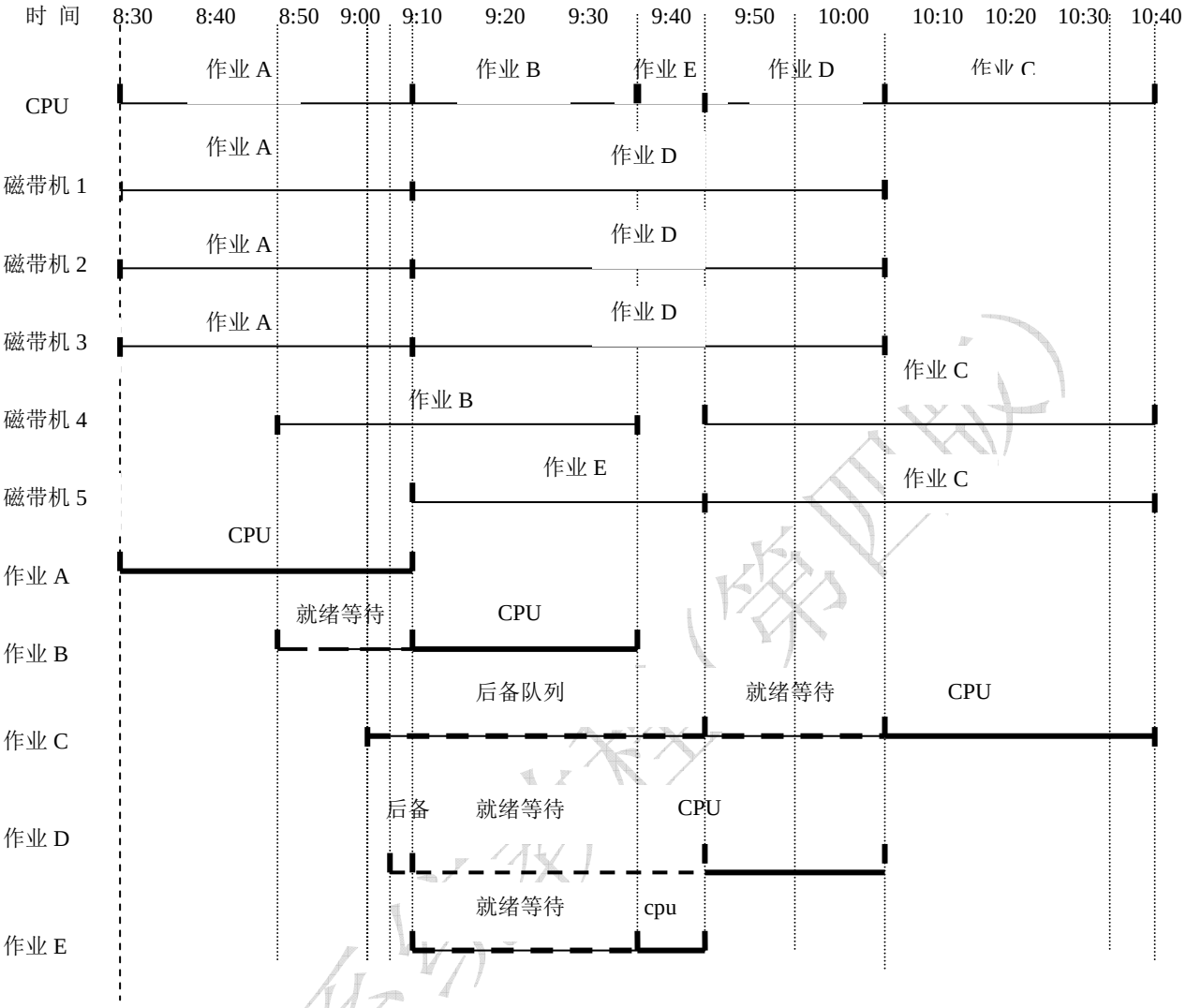
(6)9:35　　作业 B 运行结束，作业 E 投入运行。这时作业 C 因**磁带机不够**继续在后备作业队列等待。

(7)9:45　　作业 E 运行结束，作业 D 投入运行。作业 C 调入主存进就绪队列等 CPU。

(8)10:05　　作业 D 运行结束，作业 C 投入运行。

(9)10:40　　作业 C 运行结束。





作业执行次序	进输入井时间	装入主存时间	开始执行时间	执行结束时间	周转时间
作业 A	8:30	8:30	8:30	9:10	40(分)
作业 B	8:50	8:50	9:10	9:35	45
作业 E	9:10	9:10	9:35	9:45	35
作业 D	9:05	9:10	9:45	10:05	60
作业 C	9:00	9:45	10:05	10:40	100
作业平均周转时间		(40+45+35+50+100)/5=56 分钟			

30　多道批处理系统中配有一台处理器和两台外设（Il 和 I2），用户存储空间为 100MB。已知系统的作业调度及进程调度采用可抢占的高优先数调度算法，主存采用不允许移动的可变分区分配策略，设备分配按照动态分配原则。今有 4 个作业同时提交给系统，如下表所示。试求作业平均周转时间。

作业名	优先数	运行时间与顺序(分钟)	主存需求
A	7	CPU-1 分，I1-2 分，I2-2 分	50MB
B	3	CPU-3 分，I1-1 分	10MB
C	9	CPU-2 分，I1-3 分，CPU-2 分	60MB
D	4	CPU-4 分，I1-1 分	20MB

答：本题是综合性题目，考核要点是作业调度、主存分配及作业周转时间等。当 4 个作业进入系统后：

(1)按照高优先级调度算法，系统先调度作业 C。主存被 C 占有 60M，还有 40M 可用空间。系统再装入 D 和 B。

(2)同样按照高优先级算法，让 C 先运行。两分钟后 C 让出 CPU，并占用 Il。作业 D 开始在 CPU 上执行。

(3)又过去 3 分钟，作业 C 使用 Il 完毕，被唤醒后立即抢占 CPU，使作业 D 回到就绪队列等待。

(4)2 分钟后，作业 C 运行完。系统将 C 卸出主存，继而装入作业 A。因 A 的优先数较高，故立即得到运行。

(5)作业 A 运行 l 分钟后，转而使用 Il 进行 I / O。空出的 CPU 运行作业 D。

(6)1 分钟过后，作业 D 放弃 CPU，请求 I1 因不能满足而等待。作业 B 开始运行。又过去 3 分钟，B 运行完。

CPU 的使用情况如下（其中一个格代表 1 分钟）：

C	C	D	D	D	C	C	A	D	B	B	B

I1 的使用情况如下：

		C	C	C				A	A	D	B

I2 的使用情况如下：

										A	A

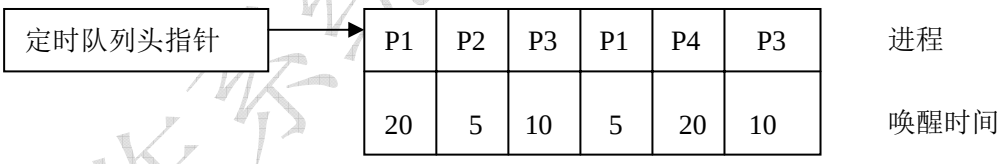
主存使用情况：

<table>
<tr><td rowspan="2">C（60）</td><td rowspan="2">C（60）</td><td rowspan="2">空</td><td>A（50）</td></tr>
<tr><td>空(10)</td></tr>
<tr><td rowspan="3">空(40)</td><td>D（20）</td><td>D（20）</td><td>D(20)</td></tr>
<tr><td>B（10）</td><td>B(10)</td><td>B(10)</td></tr>
<tr><td>空(10)</td><td>空(10)</td><td>空(10)</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td>装入 C</td><td>装 入 D、B</td><td>卸出 C</td><td>装入 A</td></tr>
</table>

作业周转时间：A=12，B=12，C=7，D=11
平均作业周转时间=（12+12+7+11）/4=42/4=10.5（分钟）

31 设计一个进程定时唤醒队列和定时唤醒处理程序：(1)说明一个等待唤醒进程入队的过程。(2)说明时钟中断时，定时唤醒处理程序的处理过程。(3)现有进程 P1 要求 20 秒后运行，经过 40 秒后再次运行；P2 要求 25 秒后运行；P3 要求 35 秒后运行，经过 35 秒后再次运行；P4 要求 60 秒后运行。试建立相应的进程定时唤醒队列。(10 分)

答：
组织如下的定时唤醒队列。



(1) 当一个需定时唤醒的进程要入队时，根据它要唤醒的时间，被扦入队列的适当位置，注意，唤醒时间按增量方式存放。
(2) 每当时钟中断时，时钟中断例程判别把队列中的第一个进程的时间量减 1，直到该值时为唤醒进程工作。同时队列中下一个进程成为队列头。

33 在 UNIX 系统中运行以下程序，最多可产生出多少进程?画出进程家属树。

```
main( ){
   fork( ); /*←pc(程序计数器)，进程 A
   fork( );
   fork( );
  } (10 分)
```

解：首先采用 fork( )创建的子进程，其程序是复制父进程的；其次，父、子进程都从调用后的那条语句开始执行。

当进程 A 执行后，派生出子进程 B，其程序及状态如下：

```
main( ) {
    fork( );
    fork( ); /*←pc(程序计数器)，进程 A */
    fork( );
  }
```

```
main( )  {
 fork( );
 fork( ); /*←pc(程序计数器)，进程 B*/
 fork( );
 }
```

当进程 A、B 执行后，各派生出子进程 C、D，其程序及状态如下：

```
main( ) {
    fork( );
    fork( );
    fork( ); /*←pc(程序计数器)，进程 A*/
  }
```

```
main( )  {
 fork( );
 fork( );
 fork( );
fork( ); /*←pc(程序计数器)，进程 B  */
    }
```

```
main( ) {
    fork( );
    fork( );
    fork( ); /*←pc(程序计数器)，进程 C*/
  }
```

```
main( )  {
 fork( );
 fork( );
 fork( );
fork( ); /*←pc(程序计数器)，进程 D */
    }
```

当进程 A、B、C、D 执行后，各派生出子进程 E、F、G、H，且所有进程的 PC 均指向程序结束处。这时进程 A 共派生出 7 个子进程。