CSI3131 - Operating Systems
Tutoring 4
CPU Scheduling - Solution

1. Define the difference between pre-emptive and cooperative scheduling.

Pre-emptive scheduling interrupts a process before the end of its CPU burst, removing the CPU and allocating it to another process. The co-operative order ensures that a process will have control of the CPU until the end of its burst of CPU.

2. What is the advantage of having different quantums for different levels of scheduling with multi-level queues.

We place in a queue with a **small quantum, processes that need more frequent service (and faster response times), for example, interactive processes. In a queue with longer quantums, processes are placed without these needs, and thus reduce the number of context switches (making computer use more efficient).**

3. Several CPU scheduling algorithms include parameters. For example, the round-robin algorithm asks for a value for the time slot. Multi-level return queues include have settings to set the number of queues, algorithms for each queue, criteria for moving processes between queues, and so on. These algorithms are really a set of algorithms (for example, all turnstiles algorithms with all possible time slice values, etc.). One set of algorithms may include another set (for example, the round-robin algorithm with the infinite time slice). What is the relationship (if it exists) between the peers of the following algorithms:

⇒ Priority and SJF **The value of the next burst serves as a priority for process.**

⇒ Multi-level Files back and FCFS The lower level of **the muli-level queues** normally **uses the FCFS.**

⇒Priority and FCFS **FCFS prioritizes processes based on time of existence in the Ready queue.**

⇒ Round-robin and SJF **None.**

4. Suppose a scheduling algorithm (at the CPU level) promotes processes that have used CPU for the least recent time. Why does this algorithm promote IO-dependent processes without starving CPU-dependent processes? Is it possible that CPU processes are starving and under what conditions?

It favours processes that depend on I/O because they have only short bursts of CPU; and CPU-dependent processes will not be starving as the IO-dependent processes often release CPU to make their I/O. If the number of processes dependent on I/O is high enough to ensure at least one I/O-dependent process in the Ready line, the CPU-dependent process will be in famine.

5. Take for granted that the operating system associates user threads with LWP (light weight processes)in a multi-to-several model. In addition, the system will allow a program to use threads in real time. Is it necessary to make a permanent association between a real-time thread and an LWP?

Yes, otherwise the real-time thread must compete with other threads for an available LWP before having access to the CPU. With a permanent association with an LWP, a real-time thread can be ordered by the core faster.
Also the OS can associate a high priority with such a thread to receive a service in real time.

6. Suppose the following processes arrive to run at the specified times. Each process rolls for the indicated burst times.

Process	Arrival time	Burst time
P1	0.0	8
P2	0.4	4
P3	1.0	1

a) How long will it take to rotate processes with the FCFS scheduling algorithm?

P1 arrives and starts running at 0.0 and finishes at 8, so its rotation time is (8-0.0) - 8 P2 arrives at 0.4, starts its run at 8 and finishes at 12, so its rotation time is (12-0.4) - 11.6 P3 arrives at 1.0, starts its execution at 12 and finishes at 13, so its rotation time is (13-1)12. Average rotation time - (8 - 11.6 - 12)/3 - 10.53

b) What will be the rotation time for processes with the shortest first scheduling algorithm - SJF (without pre-emption)?

P1 arrives and starts its run at 0.0 and finishes at 8, so its rotation time is (8-0) 8. P2 arrives at 0.4, starts running at 9 (after P3 has run), and finishes at 13, so its rotation time is (13-0.4) - 12.6.
P3 arrives at 1.0, starts its run at 8 and finishes at 9, so its rotation time is (9-1) 8. Average rotation time - (8-12.6 -8)/3 - 9.53

c) The SJF algorithm is supposed to increase performance, but notice that we chose to ride P1 at time 0 because it was not possible to know that two other processes with shorter bursts would happen soon. Calculate the average rotation time if CPU is not used for a time unit and subsequently an SJF scheduling. Remember that the *P1* and *P2* processes wait for rest time, and therefore their waiting time may increase. This algorithm could be known as the knowledge algorithm of the future.

Remember that the CPU is left idle for the first 1 time unit.
Remember that CPU is off for the first unit of time.
P1 arrives at 0.0, starts running at 6 (after P2 and P3 have run) and finishes at 14, so its rotation time is (14-0) 14.
P2 arrives at 0.4, starts its run at 2.0 (after P3 has run), and finishes at 6, so its rotation time is (6-0.4) - 5.6.
P3 arrives at 1.0, starts its run at 1 (it has the shortest burst and is the first to run) and finishes at 2, so its rotation time is (2-1) 1. Average rotation time - (14-5.6-1)/3 - 6.87

d) What will be the rotation time for processes with the shortest pre-emptive first scheduling algorithm - preemptive SJF(SRTF) shortest remaining timefirst??

P1 arrives at 0.0, begins its execution
at 0,
is pre-empted by P2 at 0.4 having 7.6 u.t. to run,
resumes only after P2 and P3 have finished their run at 5.4, and finishes
at (5.4-7.6) 13, so its rotation time is 13-0 - 13.
P2 arrives at 0.4, begins its execution at
0.4,
is pre-empted by P3 to 1 having 3.4 u.t.s to run, resumes
only after P3 has finished its run at 2, and finishes at (2-
3.4) - 5.4, so its rotation time is (5.4 - 0.4) - 5.
P3 arrives at 1.0, preempts P2 in execution and starts its execution at 1 and finishes at 2, so its rotation time is (2-1) 1.
Average rotation time -(13-5-1)/3 - 6.33
To be fair to other algorithms, the number of context switches should be added to calculated rotation times.
The pre-emptive SJF gives the largest number of context switches.

7. Examine the following CPU bursts from the three P1, P2 and P3 processes:

CPU bursts for P1: 14, 12, 17
CPU bursts for P2: 2, 2, 2, 3, 2, 2, 2, 2, 3
CPU bursts for P3: 6, 3, 8, 2, 1, 3, 4, 1, 9, 7

The three processes arrive at time 0, in order P1, P2, P3. Each burst of CPU is followed by an I/S operation that takes 6 units of time, with the exception of the last burst after which the process completes. Using provided tables, simulate the execution of the three processes to an end with the following scheduling algorithms. Ignore the time of context switching and scheduling functions (i.e. are equal to 0). a) FCFS

b) SJF (without pre-emption)
c) Pre-emptive SJF
d) Round-robin with a quantum of 5 units of time.
e) Round-robin with 5-unit quantum of time and priorities: P2-P3
f) Multi-level return file with three queues and the following settings:
    i. File 0 - 2-unit quantum of time (after which the process is moved to line 1).
    ii. File 1 - quantum of 4 time units (after which the process is moved to line 2)
    iii. File 2 - FCFS
    iv. All processes that become ready are added to lane 0.
    v. A process that arrives in lane 0 preempts any process that happens running that belongs to lane 1 or line 2.
    vi. The lane 1 processes are ordered at CPU only when line 0 is empty.
    vii. The processes in lane 2 are ordered at CPU only when line 0 and line 1 are empty.

Table for algorithms has to d:

Algorithm:				
Time	**Cpu**	**File Ready**	**Blocked file (waiting)**	**Finished**

Scheduling multi-level file back						
Time	**Cpu**	**File 0**	**File 1**	**File 2**	**Blocked**	**Finished**

FCFS				
Time	**Cpu**	**File Ready**	**Blocked file (waiting)**	**Finished**
0	**P1 (0)**	**P2(0), P3 (0)**		
14	**P2 (0)**	**P3 (0)**	**P1 (14)**	
16	**P3 (0)**		**P1(14), P2 (2)**	
20	**P3 (4)**	**P1 (14)**	**P2 (2)**	
22	**P3 (6)**	**P1(14), P2 (2)**		
22	**P1 (14)**	**P2 (2)**	**P3 (6)**	
28	**P1 (20)**	**P2(2), P3 (6)**		
34	**P2 (2)**	**P3 (6)**	**P1 (26)**	
36	**P3 (6)**		**P1(26), P2 (4)**	
39			**P1(26), P2(4), P3 (9)**	
40	**P1 (26)**		**P2(4), P3 (9)**	
42	**P1 (28)**	**P2 (4)**	**P3 (9)**	
45	**P1 (31)**	**P2(4), P3 (9)**		
57	**P2 (4)**	**P3 (9)**		**P1 (43)**
59	**P3 (9)**		**P2 (6)**	**P1 (43)**
65	**P3 (15)**	**P2 (6)**		**P1 (43)**
67	**P2 (6)**		**P3 (17)**	**P1 (43)**
70			**P3(17), P2 (9)**	**P1 (43)**
73	**P3 (17)**		**P2 (9)**	**P1 (43)**
75			**P2(9), P3 (19)**	**P1 (43)**
76	**P2 (9)**		**P3 (19)**	**P1 (43)**
78			**P3(19), P2 (11)**	**P1 (43)**
81	**P3 (19)**		**P2 (11)**	**P1 (43)**
82			**P2(11), P3 (20)**	**P1 (43)**
84	**P2 (11)**		**P3 (20)**	**P1 (43)**
86			**P3(20), P2 (13)**	**P1 (43)**
88	**P3 (20)**		**P2 (13)**	**P1 (43)**
91			**P2(13), P3 (23)**	**P1 (43)**
92	**P2 (13)**		**P3 (23)**	**P1 (43)**
94			**P3(23), P2 (15)**	**P1 (43)**
97	**P3 (23)**		**P2 (15)**	**P1 (43)**
100	**P3 (26)**	**P2 (15)**		**P1 (43)**
101	**P2 (15)**		**P3 (27)**	**P1 (43)**
104			**P3(27), P2 (18)**	**P1 (43)**
107	**P3 (27)**		**P2 (18)**	**P1 (43)**
108			**P2(18), P3 (28)**	**P1 (43)**
110	**P2 (18)**		**P3 (28)**	**P1 (43)**
112			**P3(28), P2 (20)**	**P1 (43)**
114	**P3 (28)**		**P2 (20)**	**P1 (43)**

118	P3 (32)	P2 (20)		P1 (43)
123	P2 (20)		P3 (37)	P1 (43)
125			P3(37), P2 (22)	P1 (43)
129	P3 (37)		P2 (22)	P1 (43)
131	P3 (39)	P2 (22)		P1 (43)
136	P2 (22)			P1(43), P3 (44)
138			P2 (24)	P1(43), P3 (44)
144	P2 (24)			P1(43), P3 (44)
147				P1(43), P3 (44), P2 (27)

Round-robin (quantum - 5)				
Time	Cpu	File Ready	Blocked file (waiting)	Finished
0	P1 (0)	P2(0), P3 (0)		
5	P2 (0)	P3(0), P1 (5)		
7	P3 (0)	P1 (5)	P2 (2)	
12	P1 (5)	P3 (5)	P2 (2)	
13	P1 (6)	P3(5), P2 (2)		
17	P3 (5)	P2(2), P1 (10)		
18	P2 (2)	P1 (10)	P3 (6)	
20	P1 (10)		P3(6), P2 (4)	
24	P1 (14)	P3 (6)	P2 (4)	
24	P3 (6)		P2(4), P1 (14)	
26	P3 (8)	P2 (4)	P1 (14)	
27	P2 (4)		P1(14), P3 (9)	
29			P1(14), P3 (9), P2 (6)	
30	P1 (14)		P3 (9), P2 (6)	
33	P1 (17)	P3 (9)	P2 (6)	
35	P1 (19)	P3 (9), P2 (6)		
35	P3 (9)	P2(6), P1 (19)		
40	P2 (6)	P1(19), P3 (14)		
43	P1 (19)	P3 (14)	P2 (9)	
48	P3 (14)	P1 (24)	P2 (9)	
49	P3 (15)	P1(24), P2 (9)		
51	P1 (24)	P2 (9)	P3 (17)	
53	P2 (9)		P3(17), P1 (26)	
55			P3 (17), P1(26), P2 (11)	
57	P3 (17)		P1(26), P2 (11)	
59	P3 (19)	P1 (26)	P2 (11)	
59	P1 (26)		P2(11), P3 (19)	
61	P1 (28)	P2 (11)	P3 (19)	
64	P2 (11)	P1 (31)	P3 (19)	
65	P2 (12)	P1(31), P3 (19)		
66	P1 (31)	P3 (19)	P2 (13)	
71	P3 (19)	P1 (36)	P2 (13)	
72	P3 (20)	P1(36), P2 (13)		

72	**P1 (36)**	**P2 (13)**	**P3 (20)**	
77	**P2 (13)**	**P1 (41)**	**P3 (20)**	
78	**P2 (14)**	**P1(41), P3 (20)**		
79	**P1 (41)**	**P3 (20)**	**P2 (15)**	
81	**P3 (20)**		**P2 (15)**	**P1 (43)**
84			**P2(15), P3 (23)**	**P1 (43)**
85	**P2 (15)**		**P3 (23)**	**P1 (43)**
88			**P3(23), P2 (18)**	**P1 (43)**
90	**P3 (23)**		**P2 (18)**	**P1 (43)**
94	**P3 (27)**	**P2 (18)**		**P1 (43)**
94	**P2 (18)**		**P3 (27)**	**P1 (43)**
96			**P3(27), P2 (20)**	**P1 (43)**
100	**P3 (27)**		**P2 (20)**	**P1 (43)**
101			**P2(20), P3 (28)**	**P1 (43)**
102	**P2 (20)**		**P3 (28)**	**P1 (43)**
104			**P3(28), P2 (22)**	**P1 (43)**
107	**P3 (28)**		**P2 (22)**	**P1 (43)**
110	**P3 (31)**	**P2 (22)**		**P1 (43)**
Round-robin (quantum - 5)				
Time	**Cpu**	**File Ready**	**Blocked file (waiting)**	**Finished**
112	**P2 (22)**	**P3 (33)**		**P1 (43)**
114	**P3 (33)**		**P2 (24)**	**P1 (43)**
118			**P2(24), P3 (37)**	**P1 (43)**
120	**P2 (24)**		**P3 (37)**	**P1 (43)**
123			**P3 (37)**	**P1(43), P2 (27)**
124	**P3 (37)**			**P1(43), P2 (27)**
129	**P3 (42)**			**P1(43), P2 (27)**
131				**P1(43), P2 (27), P3 (44)**

SJF (no preemption)				
Time	**Cpu**	**File Ready**	**Blocked file (waiting)**	**Finished**
0	P1 (0)	P2(0), P3 (0)		
14	P2 (0)	P3 (0)	P1 (14)	
16	P3 (0)		P1(14), P2 (2)	
20	P3 (4)	P1 (14)	P2 (2)	
22	P3 (6)	P1(14), P2 (2)		
22	P2 (2)	P1 (14)	P3 (6)	
24	P1 (14)		P3(6), P2 (4)	
28	P1 (18)	P3 (6)	P2 (4)	
30	P1 (20)	P3(6), P2 (4)		
36	P2 (4)	P3 (6)	P1 (26)	
38	P3 (6)		P1(26), P2 (6)	
41			P1(26), P2(6), P3 (9)	
42	P1 (26)		P2(6), P3 (9)	
44	P1 (28)	P2 (6)	P3 (9)	
47	P1 (31)	P2(6), P3 (9)		
59	P2 (6)	P3 (9)		P1 (43)
62	P3 (9)		P2 (9)	P1 (43)
68	P3 (15)	P2 (9)		P1 (43)
70	P2 (9)		P3 (17)	P1 (43)
72			P3(17), P2 (11)	P1 (43)
76	P3 (17)		P2 (11)	P1 (43)
78	P3 (19)	P2 (11)		P1 (43)
78	P2 (11)		P3 (19)	P1 (43)
80			P3(19), P2 (13)	P1 (43)
84	P3 (19)		P2 (13)	P1 (43)
85			P2(13), P3 (20)	P1 (43)
86	P2 (13)		P3 (20)	P1 (43)
88			P3(20), P2 (15)	P1 (43)
91	P3 (20)		P2 (15)	P1 (43)
94	P3 (23)	P2 (15)		P1 (43)
94	P2 (15)		P3 (23)	P1 (43)
97			P3(23), P2 (18)	P1 (43)
100	P3 (23)		P2 (18)	P1 (43)
103	P3 (26)	P2 (18)		P1 (43)
104	P2 (18)		P3 (27)	P1 (43)
106			P3(27), P2 (20)	P1 (43)
110	P3 (27)		P2 (20)	P1 (43)
111			P2(20), P3 (28)	P1 (43)
112	P2 (20)		P3 (28)	P1 (43)
114			P3(28), P2 (22)	P1 (43)
117	P3 (28)		P2 (22)	P1 (43)
120	P3 (31)	P2 (22)		P1 (43)
126	P2 (22)		P3 (37)	P1 (43)
128			P3(37), P2 (24)	P1 (43)
132	P3 (37)		P2 (24)	P1 (43)
134	P3 (39)	P2 (24)		P1 (43)

139	P2 (24)			P1(43), P3 (44)
142				P1(43), P3 (44), P2 (27)

Pre-emptive SJF				
Time	Cpu	File Ready	Blocked file (waiting)	Finished
0	P2 (0)	P1(0), P3 (0)		
2	P3 (0)	P1 (0)	P2 (2)	
8	P3 (6)	P1(0), P2 (2)		
8	P2 (2)	P1 (0)	P3 (6)	
10	P1 (0)		P3(6), P2 (4)	
14	P3 (6)	P1 (4)	P2 (4)	
16	P3 (8)	P1(4), P2 (4)		
17	P2 (4)	P1 (4)	P3 (9)	
19	P1 (4)		P3 (9), P2 (6)	
23	P1 (8)	P3 (9)	P2 (6)	
25	P2 (6)	P3 (9), P1 (10)		
28	P1 (10)	P3 (9)	P2 (9)	
32	P3 (9)		P2(9), P1 (14)	
34	P2 (9)	P3 (11)	P1 (14)	
36	P3 (11)		P1(14), P2 (11)	
38	P3 (13)	P1 (14)	P2 (11)	
42	P3 (17)	P1(14), P2 (11)		
42	P2 (11)	P1 (14)	P3 (17)	
44	P1 (14)		P3(17), P2 (13)	
48	P3 (17)	P1 (18)	P2 (13)	
50	P3 (19)	P1(18), P2 (13)		
50	P2 (13)	P1 (18)	P3 (19)	
52	P1 (18)		P3(19), P2 (15)	
56	P3 (19)	P1 (22)	P2 (15)	
57	P1 (22)		P2(15), P3 (20)	
58	P2 (15)	P1 (23)	P3 (20)	
61	P1 (23)		P3(20), P2 (18)	
63	P1 (25)	P3 (20)	P2 (18)	
64	P3 (20)		P2(18), P1 (26)	
67	P3 (23)	P2 (18)	P1 (26)	
67	P2 (18)		P1(26), P3 (23)	
69			P1(26), P3 (23), P2 (20)	
70	P1 (26)		P3(23), P2 (20)	
73	P3 (23)	P1 (29)	P2 (20)	
75	P2 (20)	P1(29), P3 (25)		
77	P3 (25)	P1 (29)	P2 (22)	
79	P1 (29)		P2(22), P3 (27)	
83	P2 (22)	P1 (33)	P3 (27)	
85	P2 (24)	P1(33), P3 (27)		
85	P3 (27)	P1 (33)	P2 (24)	
86	P1 (33)		P2(24), P3 (28)	
91	P2 (24)	P1 (38)	P3 (28)	

92	P2 (25)	P1(38), P3 (28)		
94	P1 (38)	P3 (28)		P2 (27)
99	P3 (28)			P2(27), P1 (43)
108			P3 (37)	P2(27), P1 (43)
114	P3 (37)			P2(27), P1 (43)
121				P2(27), P1 (43), P3 (44)

Round-robin (quantum-5) with priority (P2-P3 - P1)				
Time	Cpu	File Ready	Blocked file (waiting)	Finished
0	P2 (0)	P1(0), P3 (0)		
2	P3 (0)	P1 (0)	P2 (2)	
7	P3 (5)	P1 (0)	P2 (2)	
8	P3 (6)	P1(0), P2 (2)		
8	P2 (2)	P1 (0)	P3 (6)	
10	P1 (0)		P3(6), P2 (4)	
14	P3 (6)	P1 (4)	P2 (4)	
16	P3 (8)	P1(4), P2 (4)		
17	P2 (4)	P1 (4)	P3 (9)	
19	P1 (4)		P3 (9), P2 (6)	
23	P3 (9)	P1 (8)	P2 (6)	
25	P3 (11)	P1(8), P2 (6)		
28	P2 (6)	P1(8), P3 (14)		
31	P3 (14)	P1 (8)	P2 (9)	
34	P1 (8)		P2(9), P3 (17)	
37	P2 (9)	P1 (11)	P3 (17)	
39	P1 (11)		P3(17), P2 (11)	
40	P3 (17)	P1 (12)	P2 (11)	
42	P1 (12)		P2(11), P3 (19)	
44			P2(11), P3 (19), P1 (14)	
45	P2 (11)		P3(19), P1 (14)	
47			P3(19), P1 (14), P2 (13)	
48	P3 (19)		P1(14), P2 (13)	
49			P1(14), P2 (13), P3 (20)	
50	P1 (14)		P2(13), P3 (20)	
53	P2 (13)	P1 (17)	P3 (20)	
55	P2 (15)	P1(17), P3 (20)		
55	P3 (20)	P1 (17)	P2 (15)	
58	P1 (17)		P2(15), P3 (23)	
61	P2 (15)	P1 (20)	P3 (23)	
64	P2 (18)	P1(20), P3 (23)		
64	P3 (23)	P1 (20)	P2 (18)	
68	P1 (20)		P2(18), P3 (27)	
70	P2 (18)	P1 (22)	P3 (27)	
72	P1 (22)		P3(27), P2 (20)	
74	P3 (27)	P1 (24)	P2 (20)	
75	P1 (24)		P2(20), P3 (28)	
77			P2(20), P3 (28), P1 (26)	

78	P2 (20)		P3(28), P1 (26)	
80			P3 (28), P1(26), P2 (22)	
81	P3 (28)		P1(26), P2 (22)	
83	P3 (30)	P1 (26)	P2 (22)	
86	P3 (33)	P1(26), P2 (22)		
86	P2 (22)	P1(26), P3 (33)		
88	P3 (33)	P1 (26)	P2 (24)	
92	P1 (26)		P2(24), P3 (37)	
94	P2 (24)	P1 (28)	P3 (37)	
97	P1 (28)		P3 (37)	P2 (27)
98	P3 (37)	P1 (29)		P2 (27)
103	P3 (42)	P1 (29)		P2 (27)
105	P1 (29)			P2(27), P3 (44)
110	P1 (34)			P2(27), P3 (44)
115	P1 (39)			P2(27), P3 (44)
119				P2(27), P3 (44), P1 (43)

Multi-level file back						
Time	Cpu	File 0	File 1	File 2	Blocked	Finished
0	P1 (0)	P2(0), P3 (0)				
2	P2 (0)	P3 (0)	P1 (2)			
4	P3 (0)		P1 (2)		P2 (2)	
6	P1 (2)		P3 (2)		P2 (2)	
10	P2 (2)		P3 (2)	P1 (6)		
12	P3 (2)			P1 (6)	P2 (4)	
16	P1 (6)				P2(4), P3 (6)	
18	P2 (4)			P1 (8)	P3 (6)	
20	P1 (8)				P3 (6), P2 (6)	
22	P3 (6)			P1 (10)	P2 (6)	
24	P3 (8)			P1 (10)	P2 (6)	
25	P1 (10)				P2(6), P3 (9)	
26	P2 (6)			P1 (11)	P3 (9)	
28	P2 (8)			P1 (11)	P3 (9)	
29	P1 (11)				P3(9), P2 (9)	
31	P3 (9)			P1 (13)	P2 (9)	
33	P3 (11)			P1 (13)	P2 (9)	
35	P2 (9)		P3 (13)	P1 (13)		
37	P3 (13)			P1 (13)	P2 (11)	
39	P1 (13)			P3 (15)	P2 (11)	
40	P3 (15)				P2(11), P1 (14)	
42					P2(11), P1 (14), P3 (17)	
43	P2 (11)				P1(14), P3 (17)	
45					P1(14), P3 (17), P2 (13)	
46	P1 (14)				P3(17), P2 (13)	
48	P1 (16)	P3 (17)			P2 (13)	
48	P3 (17)		P1 (16)		P2 (13)	
50	P1 (16)				P2(13), P3 (19)	

51	P2 (13)		P1 (17)		P3 (19)	
53	P1 (17)				P3(19), P2 (15)	
56	P3 (19)			P1 (20)	P2 (15)	
57	P1 (20)				P2(15), P3 (20)	
59	P2 (15)			P1 (22)	P3 (20)	
61	P2 (17)			P1 (22)	P3 (20)	
62	P1 (22)				P3(20), P2 (18)	
63	P3 (20)			P1 (23)	P2 (18)	
65	P3 (22)			P1 (23)	P2 (18)	
66	P1 (23)				P2(18), P3 (23)	
68	P2 (18)			P1 (25)	P3 (23)	
70	P1 (25)				P3(23), P2 (20)	
71					P3(23), P2(20), P1 (26)	
72	P3 (23)				P2(20), P1 (26)	
74	P3 (25)				P2(20), P1 (26)	
76	P2 (20)				P1(26), P3 (27)	
77	P2 (21)	P1 (26)			P3 (27)	
78	P1 (26)				P3(27), P2 (22)	
80	P1 (28)				P3(27), P2 (22)	
82	P3 (27)		P1 (30)		P2 (22)	
83	P1 (30)				P2(22), P3 (28)	
84	P2 (22)		P1 (31)		P3 (28)	
86	P1 (31)				P3(28), P2 (24)	
87	P1 (32)				P3(28), P2 (24)	
89	P3 (28)			P1 (34)	P2 (24)	
91	P3 (30)			P1 (34)	P2 (24)	
92	P2 (24)		P3 (31)	P1 (34)		
94	P3 (31)		P2 (26)	P1 (34)		
Multi-level file back						
Time	**Cpu**	**File 0**	**File 1**	**File 2**	**Blocked**	**Finished**
97	P2 (26)			P1(34), P3 (34)		
98	P1 (34)			P3 (34)		P2 (27)
107	P3 (34)					P2(27), P1 (43)
110					P3 (37)	P2(27), P1 (43)
116	P3 (37)					P2(27), P1 (43)
118	P3 (39)					P2(27), P1 (43)
122	P3 (43)					P2(27), P1 (43)
123						P2(27), P1 (43), P3 (44)