Memory Management

Paging + Preemptive SJF

Consider the following assumptions

- Each process needs 4 pages to execute
- The OS has 10 frames available, with no backing store
- The free-frame list is treated as a queue and its initial contents are:
  3, 8, 0, 2, 4, 6, 1, 9, 7, 5
- Memory allocation is done using paging and the CPU scheduling scheme is pre-emptive shortest-job first

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {3,8,0,2,4,6,1,9,7,5}

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {4,6,1,9,7,5}
P1: {3,8,0,2}

0	P1
1	
2	P1
3	P1
4	
5	
6	
7	
8	P1
9	

Time: 0
RQ = []
* P1 is in the RQ before being executed

P1(7)
0 - - - - - - -

P1 is allocated the first 4 free frames

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {7,5}
P1: {3,8,0,2}
P2: {4,6,1,9}

0	P1
1	P2
2	P1
3	P1
4	P2
5	
6	P2
7	
8	P1
9	P2

Time: 1
RQ = [P1(6)]
* P2 is in the RQ before context switching with P1

P1(6) P2(5)
0 - - - - - - - 1 - - - - - - -

P2 is allocated the first 4 free frames

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {7,5}
P1: {3,8,0,2}
P2: {4,6,1,9}

Frame	
0	P1
1	P2
2	P1
3	P1
4	P2
5	
6	P2
7	
8	P1
9	P2

Time: 2
RQ = [P3(4), P1(6)]

P1(6) P2(4)
0 - - - - - - - 1 - - - - - - -

P2 & P3 have the same CPU burst, but P2 is already being executed

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {7,5}
P1: {3,8,0,2}
P2: {4,6,1,9}

Frame	
0	P1
1	P2
2	P1
3	P1
4	P2
5	
6	P2
7	
8	P1
9	P2

Time: 3
RQ = [P4(2), P3(4), P1(6)]

P1(6)   P2(3)
0 - - - - - - - 1 - - - - - - -

P4 has lower remaining CPU burst compared to P2, but there are not enough available frames

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {7,5}
P1: {3,8,0,2}
P2: {4,6,1,9}

0	P1
1	P2
2	P1
3	P1
4	P2
5	
6	P2
7	
8	P1
9	P2

Time: 4
RQ = [P4(2), P5(3), P3(4), P1(6)]

P1(6) P2(2)
0 - - - - - - - 1 - - - - - - -

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {7,5}

P1: {3,8,0,2}

P2: {4,6,1,9}

0	P1
1	P2
2	P1
3	P1
4	P2
5	
6	P2
7	
8	P1
9	P2

Time: 5

RQ = [P4(2), P5(3), P3(4), P1(6)]

P1(6) P2(1)

0 - - - - - - - 1 - - - - - - -

Paging + Preemptive SJF

Process	**Arrival Time**	**CPU Burst**
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {1,9}
P1: {3,8,0,2}
P4: {7,5,4,6}

0	P1
1	
2	P1
3	P1
4	P4
5	P4
6	P4
7	P4
8	P1
9	

Time: 6
RQ = [P5(3), P3(4), P1(6)]

P1(6) P2(0) P4(2)
0 - - - - - - - 1 - - - - - - - 6 - - - - - - -

The frames used by P2 are reclaimed and added to the end of the free-frames list

P4 is allocated the first 4 free frames

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {1,9}
P1: {3,8,0,2}
P4: {7,5,4,6}

0	P1
1	
2	P1
3	P1
4	P4
5	P4
6	P4
7	P4
8	P1
9	

Time: 7
RQ = [P5(3), P3(4), P1(6)]

P1(6) P2(0) P4(1)
0 - - - - - - - 1 - - - - - - - 6 - - - - - - -

Paging + Preemptive SJF

Process	**Arrival Time**	**CPU Burst**
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {4,6}
P1: {3,8,0,2}
P5: {1,9,7,5}

0	P1
1	P5
2	P1
3	P1
4	
5	P5
6	
7	P5
8	P1
9	P5

Time: 8
RQ = [P3(4), P1(6)]

P1(6) P2(0) P4(0) P5(3)
0 - - - - - - - 1 - - - - - - - 6 - - - - - - - 8 - - - - - - -

The frames used by P4 are reclaimed and added to the end of the free-frames list

P5 is allocated the first 4 free frames

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {4,6}
P1: {3,8,0,2}
P5: {1,9,7,5}

0	P1
1	P5
2	P1
3	P1
4	
5	P5
6	
7	P5
8	P1
9	P5

Time: 9
RQ = [P3(4), P1(6)]

P1(6) P2(0) P4(0) P5(2)
0 - - - - - - 1 - - - - - - 6 - - - - - - 8 - - - - - -

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {4,6}
P1: {3,8,0,2}
P5: {1,9,7,5}

Frame	
0	P1
1	P5
2	P1
3	P1
4	
5	P5
6	
7	P5
8	P1
9	P5

Time: 10
RQ = [P3(4), P1(6)]

P1(6) P2(0) P4(0) P5(1)
0 - - - - - - 1 - - - - - - 6 - - - - - - 8 - - - - - -

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {7,5}
P1: {3,8,0,2}
P3: {4,6,1,9}

Frame	
0	P1
1	P3
2	P1
3	P1
4	P3
5	
6	P3
7	
8	P1
9	P3

Time: 11
RQ = [P1(6)]

P1(6) P2(0) P4(0) P5(0) P3(4)
0 - - - - - - - 1 - - - - - - - 6 - - - - - - - 8 - - - - - - - 11 - - - - - - -

The frames used by P5 are reclaimed and added to the end of the free-frames list

P3 is allocated the first 4 free frames

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {7,5}
P1: {3,8,0,2}
P3: {4,6,1,9}

0	P1
1	P3
2	P1
3	P1
4	P3
5	
6	P3
7	
8	P1
9	P3

Time: 12
RQ = [P1(6)]

P1(6) P2(0) P4(0) P5(0) P3(3)
0 - - - - - - - 1 - - - - - - - 6 - - - - - - - 8 - - - - - - - 11 - - - - - - -

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {7,5}
P1: {3,8,0,2}
P3: {4,6,1,9}

0	P1
1	P3
2	P1
3	P1
4	P3
5	
6	P3
7	
8	P1
9	P3

Time: 13
RQ = [P1(6)]

P1(6) P2(0) P4(0) P5(0) P3(2)
0 - - - - - - - 1 - - - - - - - 6 - - - - - - - 8 - - - - - - - 11 - - - - - - -

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {7,5}
P1: {3,8,0,2}
P3: {4,6,1,9}

Frame	
0	P1
1	P3
2	P1
3	P1
4	P3
5	
6	P3
7	
8	P1
9	P3

Time: 14
RQ = [P1(6)]

P1(6) P2(0) P4(0) P5(0) P3(1)
0 - - - - - - - 1 - - - - - - - 6 - - - - - - - 8 - - - - - - - 11 - - - - - - -

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {7,5,4,6,1,9}
P1: {3,8,0,2}

0	P1
1	
2	P1
3	P1
4	
5	
6	
7	
8	P1
9	

Time: 15
RQ = []

P1(6) P2(0) P4(0) P5(0) P3(0)
0 - - - - - - - 1 - - - - - - - 6 - - - - - - - 8 - - - - - - - 11 - - - - - - - 15

P1(6)
15 - - - - - - -

The frames used by P3 are reclaimed and added to the end of the free-frames list

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {7,5,4,6,1,9}
P1: {3,8,0,2}

0	P1
1	
2	P1
3	P1
4	
5	
6	
7	
8	P1
9	

Time: 16
RQ = []

P1(6) P2(0) P4(0) P5(0) P3(0)
0 - - - - - - - 1 - - - - - - - 6 - - - - - - - 8 - - - - - - - 11 - - - - - - - 15

P1(5)
15 - - - - - - -

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {7,5,4,6,1,9}
P1: {3,8,0,2}

Frame	
0	P1
1	
2	P1
3	P1
4	
5	
6	
7	
8	P1
9	

Time: 17
RQ = []

P1(6) P2(0) P4(0) P5(0) P3(0)
0 - - - - - - - 1 - - - - - - - 6 - - - - - - - 8 - - - - - - - 11 - - - - - - - 15

P1(4)
15 - - - - - - -

Paging + Preemptive SJF

Process	**Arrival Time**	**CPU Burst**
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {7,5,4,6,1,9}
P1: {3,8,0,2}

0	P1
1	
2	P1
3	P1
4	
5	
6	
7	
8	P1
9	

Time: 18
RQ = []

P1(6) P2(0) P4(0) P5(0) P3(0)
0 - - - - - - - 1 - - - - - - - 6 - - - - - - - 8 - - - - - - - 11 - - - - - - - 15

P1(3)
15 - - - - - - -

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {7,5,4,6,1,9}

P1: {3,8,0,2}

0	P1
1	
2	P1
3	P1
4	
5	
6	
7	
8	P1
9	

Time: 19

RQ = []

P1(6) P2(0) P4(0) P5(0) P3(0)

0 - - - - - - - 1 - - - - - - - 6 - - - - - - - 8 - - - - - - - 11 - - - - - - - 15

P1(2)

15 - - - - - - -

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {7,5,4,6,1,9}
P1: {3,8,0,2}

0	P1
1	
2	P1
3	P1
4	
5	
6	
7	
8	P1
9	

Time: 20
RQ = []

P1(6) P2(0) P4(0) P5(0) P3(0)
0 - - - - - - - 1 - - - - - - - 6 - - - - - - - 8 - - - - - - - 11 - - - - - - - 15

P1(1)
15 - - - - - - -

Paging + Preemptive SJF

Process	Arrival Time	CPU Burst
P1	0	7
P2	1	5
P3	2	4
P4	3	2
P5	4	3

4 pages per process

Free frames = {7,5,4,6,1,9,3,8,0,2}

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Time: 21
RQ = []

P1(6) P2(0) P4(0) P5(0) P3(0)
0 - - - - - - - 1 - - - - - - - 6 - - - - - - - 8 - - - - - - - 11 - - - - - - - 15

P1(0)
15 - - - - - - - 21

The frames used by P1 are reclaimed and added to the end of the free-frames list

Finished

Demand Paging + Round Robin

Consider the following assumptions

- Each process needs 5 pages to execute
- The OS has 12 frames available
- The free-frame list is treated as a queue and its initial contents are:
  6, 2, 8, 0, 4, 10, 5, 11, 3, 1, 7, 9
- Pages are executed in the same order as their IDs
- Backing store is enabled. Page replacement is performed on the oldest process (pages are evicted in the same order as their IDs)
- Memory allocation is done using demand paging and the CPU scheduling scheme is round robin with time slice = 3

Demand Paging + Round Robin

Process	Arrival Time	CPU Burst
P1	0	6
P2	1	4
P3	2	5
P4	3	3
P5	4	2

5 pages per process
Time slice = 3

Free frames = {6,2,8,0,4,10,5,11,3,1,7,9}

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

Backing Store

Demand Paging + Round Robin

Process	Arrival Time	CPU Burst
P1	0	6
P2	1	4
P3	2	5
P4	3	3
P5	4	2

5 pages per process
Time slice = 3

Free frames = {10,5,11,3,1,7,9}
P1: {0:6, 1:2, 2:8, 3:0, 4:4}

allocated frame
process page

Frame	
0	P1[3]
1	
2	P1[1]
3	
4	P1[4]
5	
6	P1[0]
7	
8	P1[2]
9	
10	
11	

Backing Store

Time: 0
RQ = []
* P1 is in the RQ before being executed

P1(6)
0 - - - - - -

P1 pages are allocated the first 5 free frames

Demand Paging + Round Robin

Process	**Arrival Time**	**CPU Burst**
P1	0	6
P2	1	4
P3	2	5
P4	3	3
P5	4	2

5 pages per process
Time slice = 3

Free frames = {10,5,11,3,1,7,9}
P1: {0:6, 1:2, 2:8, 3:0, 4:4}

0	P1[3]
1	
2	P1[1]
3	
4	P1[4]
5	
6	P1[0]
7	
8	P1[2]
9	
10	
11	

Backing Store

Time: 1
RQ = [P2(4)]

P1(5)
0 - - - - - -

Demand Paging + Round Robin

Process	Arrival Time	CPU Burst
P1	0	6
P2	1	4
P3	2	5
P4	3	3
P5	4	2

5 pages per process
Time slice = 3

Free frames = {10,5,11,3,1,7,9}
P1: {0:6, 1:2, 2:8, 3:0, 4:4}

0	P1[3]
1	
2	P1[1]
3	
4	P1[4]
5	
6	P1[0]
7	
8	P1[2]
9	
10	
11	

Backing Store

Time: 2
RQ = [P2(4), P3(5)]

P1(4)
0 - - - - - -

Demand Paging + Round Robin

Process	Arrival Time	CPU Burst
P1	0	6
P2	1	4
P3	2	5
P4	3	3
P5	4	2

5 pages per process
Time slice = 3

Free frames = {7,9}
P1: {0:6, 1:2, 2:8, 3:0, 4:4}
P2: {0:10, 1:5, 2:11, 3:3, 4:1}

0	P1[3]
1	P2[4]
2	P1[1]
3	P2[3]
4	P1[4]
5	P2[1]
6	P1[0]
7	
8	P1[2]
9	
10	P2[0]
11	P2[2]

Backing Store

Time: 3
RQ = [P3(5), P1(3), P4(3)]
* P2 context switches with P1 before P4 arrives

```
  P1(3)     P2(4)
0 - - - - - - - 3 - - - - - - -
```

P2 pages are allocated the first 5 free frames

Demand Paging + Round Robin

Process	Arrival Time	CPU Burst
P1	0	6
P2	1	4
P3	2	5
P4	3	3
P5	4	2

5 pages per process
Time slice = 3

Free frames = {7,9}
P1: {0:6, 1:2, 2:8, 3:0, 4:4}
P2: {0:10, 1:5, 2:11, 3:3, 4:1}

Frame	
0	P1[3]
1	P2[4]
2	P1[1]
3	P2[3]
4	P1[4]
5	P2[1]
6	P1[0]
7	
8	P1[2]
9	
10	P2[0]
11	P2[2]

Backing Store

Time: 4
RQ = [P3(5), P1(3), P4(3), P5(2)]

```
   P1(3)     P2(3)
0 - - - - - - 3 - - - - - -
```

Demand Paging + Round Robin

Process	Arrival Time	CPU Burst
P1	0	6
P2	1	4
P3	2	5
P4	3	3
P5	4	2

5 pages per process
Time slice = 3

Free frames = {7,9}
P1: {0:6, 1:2, 2:8, 3:0, 4:4}
P2: {0:10, 1:5, 2:11, 3:3, 4:1}

0	P1[3]
1	P2[4]
2	P1[1]
3	P2[3]
4	P1[4]
5	P2[1]
6	P1[0]
7	
8	P1[2]
9	
10	P2[0]
11	P2[2]

Backing Store

Time: 5
RQ = [P3(5), P1(3), P4(3), P5(2)]

P1(3) P2(2)
0 - - - - - - - 3 - - - - - - -

Demand Paging + Round Robin

Process	Arrival Time	CPU Burst
P1	0	6
P2	1	4
P3	2	5
P4	3	3
P5	4	2

5 pages per process
Time slice = 3

Free frames = {}
P1: {3:0, 4:4}
P2: {0:10, 1:5, 2:11, 3:3, 4:1}
P3: {0:7, 1:9, 2:6, 3:2, 4:8}

0	P1[3]
1	P2[4]
2	P3[3]
3	P2[3]
4	P1[4]
5	P2[1]
6	P3[2]
7	P3[0]
8	P3[4]
9	P3[1]
10	P2[0]
11	P2[2]

Backing Store
P1[0], P1[1], P1[2]

Time: 6
RQ = [P1(3), P4(3), P5(2), P2(1)]

P1(3) P2(1) P3(5)
0 - - - - - - - 3 - - - - - - - 6 - - - - - - -

P3 pages are allocated the first 5 free frames
- Pages 0 and 1 are allocated to frames 7 and 9
- Page fault is triggered when allocating for 2, 3, 4

P1 pages 0, 1, 2 are swapped out
- Frames 6, 2, 8 are freed
- P3 pages 2, 3, 4 are allocated to the free frames

Demand Paging + Round Robin

Process	Arrival Time	CPU Burst
P1	0	6
P2	1	4
P3	2	5
P4	3	3
P5	4	2

5 pages per process
Time slice = 3

Free frames = {}
P1: {0:10, 1:5, 2:11, 3:0, 4:4}
P2: {3:3, 4:1}
P3: {0:7, 1:9, 2:6, 3:2, 4:8}

0	P1[3]
1	P2[4]
2	P3[3]
3	P2[3]
4	P1[4]
5	P1[1]
6	P3[2]
7	P3[0]
8	P3[4]
9	P3[1]
10	P1[0]
11	P1[2]

Backing Store
P2[0], P2[1], P2[2]

Time: 9
RQ = [P4(3), P5(2), P2(1), P3(2)]

P1(3) P2(1) P3(2) P1(3)
0 - - - - - - - 3 - - - - - - - 6 - - - - - - - 9 - - - - - - -

Fast forward to the next time slice (time +3)

Page fault is triggered for P1 pages 0, 1, 2
- P2 pages 0, 1, 2 are swapped out
- P1 pages 0, 1, 2 are swapped in

Demand Paging + Round Robin

Process	Arrival Time	CPU Burst
P1	0	6
P2	1	4
P3	2	5
P4	3	3
P5	4	2

5 pages per process
Time slice = 3

Free frames = {}
P2: {3:3, 4:1}
P3: {0:7, 1:9, 2:6, 3:2, 4:8}
P4: {0:10, 1:5, 2:11, 3:0, 4:4}

0	P4[3]
1	P2[4]
2	P3[3]
3	P2[3]
4	P4[4]
5	P4[1]
6	P3[2]
7	P3[0]
8	P3[4]
9	P3[1]
10	P4[0]
11	P4[2]

Backing Store
P2[0], P2[1], P2[2]

Time: 12
RQ = [P5(2), P2(1), P3(2)]

P1(3) P2(1) P3(2) P1(0) P4(3)
0 - - - - - - - 3 - - - - - - - 6 - - - - - - - 9 - - - - - - - 12 - - - - - - -

Fast forward to the next time slice (time +3)

The frames used by P1 are reclaimed and added to the end of the free-frames list

P4 pages are allocated the first 5 free frames

Demand Paging + Round Robin

Process	Arrival Time	CPU Burst
P1	0	6
P2	1	4
P3	2	5
P4	3	3
P5	4	2

5 pages per process
Time slice = 3

Free frames = {}
P2: {3:3, 4:1}
P3: {0:7, 1:9, 2:6, 3:2, 4:8}
P5: {0:10, 1:5, 2:11, 3:0, 4:4}

0	P5[3]
1	P2[4]
2	P3[3]
3	P2[3]
4	P5[4]
5	P5[1]
6	P3[2]
7	P3[0]
8	P3[4]
9	P3[1]
10	P5[0]
11	P5[2]

Backing Store
P2[0], P2[1], P2[2]

Time: 15
RQ = [P2(1), P3(2)]

P1(3) P2(1) P3(2) P1(0) P4(0)
0 - - - - - - - 3 - - - - - - - 6 - - - - - - - 9 - - - - - - - 12 - - - - - - - 15

P5(2)
15 - - - - - - -

Fast forward to the next time slice (time +3)

The frames used by P4 are reclaimed and added to the end of the free-frames list

P5 pages are allocated the first 5 free frames

Demand Paging + Round Robin

Process	Arrival Time	CPU Burst
P1	0	6
P2	1	4
P3	2	5
P4	3	3
P5	4	2

5 pages per process
Time slice = 3

Free frames = {0,4}
P2: {0:10, 1:5, 2:11, 3:3, 4:1}
P3: {0:7, 1:9, 2:6, 3:2, 4:8}

0	
1	P2[4]
2	P3[3]
3	P2[3]
4	
5	P2[1]
6	P3[2]
7	P3[0]
8	P3[4]
9	P3[1]
10	P2[0]
11	P2[2]

Backing Store

Time: 17
RQ = [P3(2)]

P1(3) P2(1) P3(2) P1(0) P4(0)
0 - - - - - - - 3 - - - - - - - 6 - - - - - - - 9 - - - - - - - 12 - - - - - - - 15

P5(0) P2(1)
15 - - - - - - - 17 - - - - - - -

Fast forward to the completion time of P5

The frames used by P5 are reclaimed and added to the end of the free-frames list

P2 pages 0, 1, 2 are swapped in

Demand Paging + Round Robin

Process	Arrival Time	CPU Burst
P1	0	6
P2	1	4
P3	2	5
P4	3	3
P5	4	2

5 pages per process
Time slice = 3

Free frames = {0,4,10,5,11,3,1}
P3: {0:7, 1:9, 2:6, 3:2, 4:8}

0	
1	
2	P3[3]
3	
4	
5	
6	P3[2]
7	P3[0]
8	P3[4]
9	P3[1]
10	
11	

Backing Store

Time: 18
RQ = []

P1(3) P2(1) P3(2) P1(0) P4(0)
0 - - - - - - - 3 - - - - - - - 6 - - - - - - - 9 - - - - - - - 12 - - - - - - - 15

P5(0) P2(0) P3(2)
15 - - - - - - - 17 - - - - - - - 18 - - - - - - -

The frames used by P2 are reclaimed and added to the end of the free-frames list

Demand Paging + Round Robin

Process	Arrival Time	CPU Burst
P1	0	6
P2	1	4
P3	2	5
P4	3	3
P5	4	2

5 pages per process
Time slice = 3

Free frames = {0,4,10,5,11,3,1,7,9,6,2,8}

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

Backing Store

Time: 20
RQ = []

P1(3) P2(1) P3(2) P1(0) P4(0)
0 - - - - - - - 3 - - - - - - - 6 - - - - - - - 9 - - - - - - - 12 - - - - - - - 15

P5(0) P2(0) P3(0)
15 - - - - - - - 17 - - - - - - - 18 - - - - - - - 20

Fast forward to the completion time of P3

The frames used by P3 are reclaimed and added to the end of the free-frames list

Finished