OS Assignment III

1. Transfer rate of hard disk=60 MB/s. Consider the following segment table:

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

What are the physical addresses for the following logical addresses:

a. 0, 430

219 + 430 = 649

b. 1, 10

2300 + 10 = 2310

c. 2, 500

Illegal address since size of segment 2 is 100 and the offset in logical address is 500.

d. 3, 400

1327 + 400 = 1727

e. 4, 112

Illegal address since size of segment 4 is 96 and the offset in logical address is 112.

2. **How logical address is divided in Pentium paging with neat diagram. What is the role of CR3 register?**

The Pentium architecture allows a segment to be as large as 4 GB, and the maximum number of segments per process is 16K.

The logical space of addresses is divided into two partitions.

The first partition consists of up to 8K segments that are private to that process. The second partition consists of up to 8K segments that are shared among the processes.

Information about the first partition is kept in the local descriptor table (LDT), information about the second is kept in the global descriptor table (GDT). Each entry in the LDT and GDT consists of an 8 byte segment descriptor with detailed information about a particular segment including the base location and limit of the segment.

The logical address is a pair (selector, offset), where the selector is a 16-bit number:

s	g	p
13	1	2

in which s designates the segment number, g indicates whether the segment is in the GDT or LDT, and p deals with protection. The offset is a 32-bit number specifying the location of the byte (or word) within the segment in question.

The Pentium architecture allows a page size of either 4 KB or 4 MB. For 4-KB pages, the Pentium uses a two-level paging scheme in which the division of the 32-bit linear address is as follows:

page number		page offset
p_1	p_2	d
10	10	12

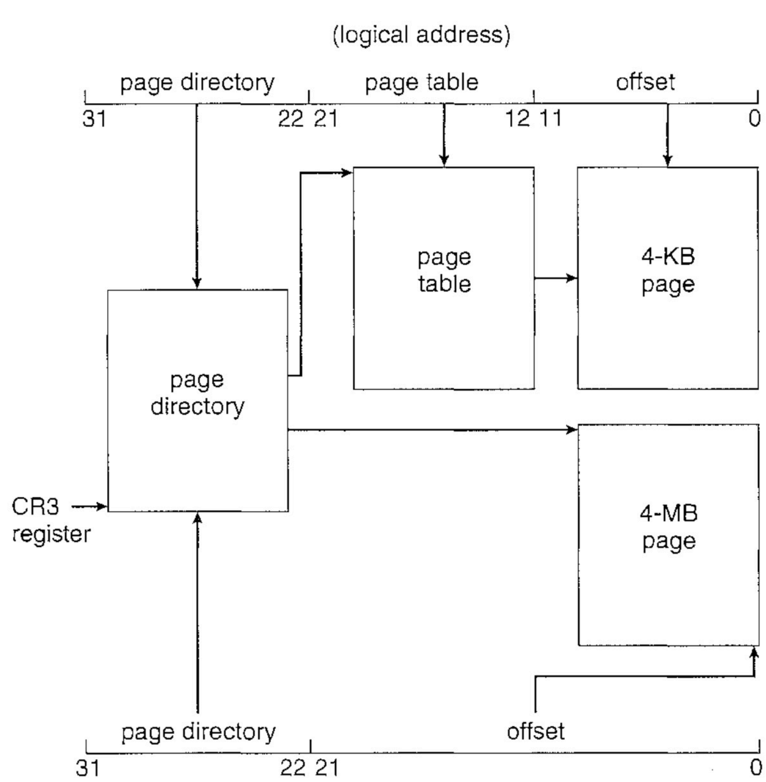
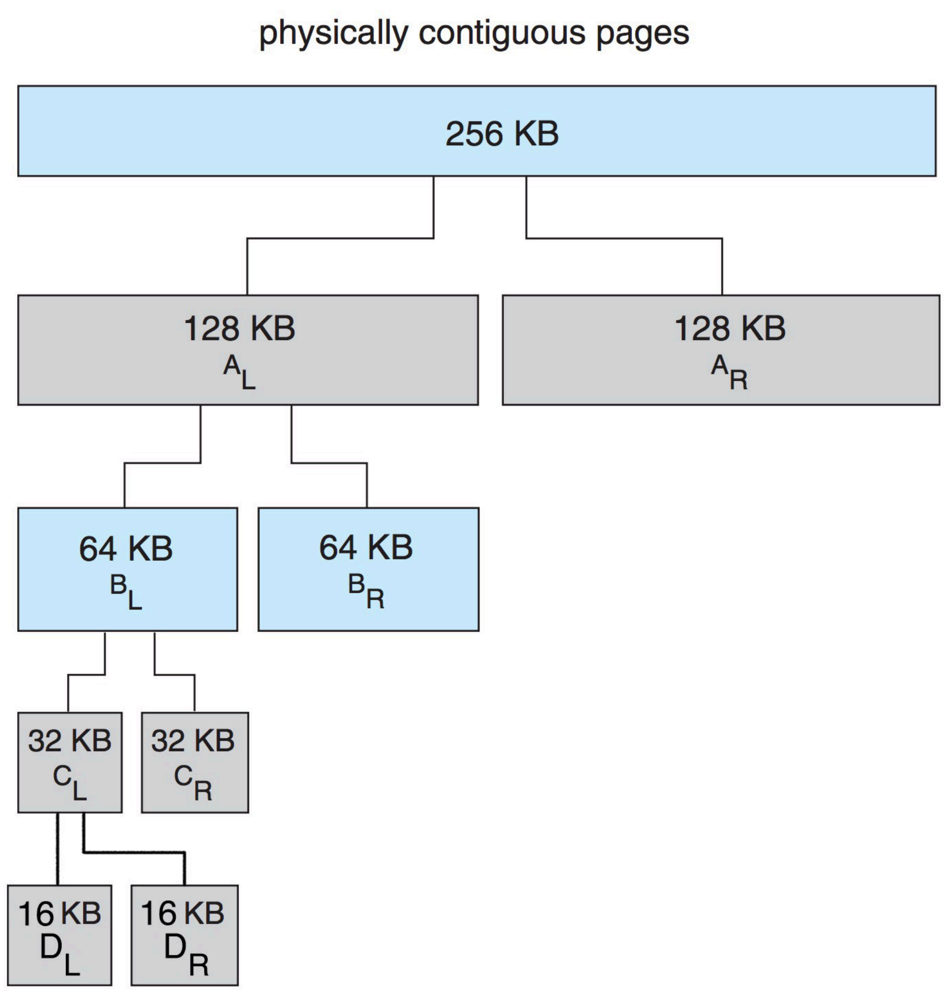


Figure 8.23 Paging in the Pentium architecture.

3. **What is the advantage of buddy system? Assume the size of memory segment is initially 256KB and the kernel request is 9KB. How power of 2 allocator works for this kernel request? Show diagrammatically.**

The buddy system allocates memory from a fixed-size segment consisting of physically contiguous pages. Memory is allocated from this segment using a **power-of-2 allocator**, which satisfies requests in units sized as a power of 2 (4 KB, 8 KB, 16 KB, and so forth).

An advantage of the buddy system is how quickly adjacent buddies can be combined to form larger segments using a technique known as **coalescing**.



The segment is initially divided into two **buddies** — which we will call A_L and A_R — each 128 KB in size. One of these buddies is further divided into two 64-KB buddies— B_L and B_R. One of these buddies is further divided into two 32-KB buddies— C_L and C_R. However, the next-highest power of 2 from 9 KB is 16 KB so either C_L or C_R is again divided into two 16-KB buddies, D_L and D_R. Now D_L is allocated to the 9 KB request.

4. **Use following page replacement algorithm and find out number of page faults for the following page reference string. Consider 3 frames.**

```
9, 5, 4, 2, 5, 8, 9, 1, 2, 4, 5, 5, 3, 8, 2, 5, 3, 2, 0
```

a. **LRU**

	9	5	4	2	5	8	9	1	2	4	5	5	3	8	2	5	3	2	0
-	9	9	9	2	2	2	9	9	9	4	4	4	4	8	8	8	3	3	3
-		5	5	5	5	5	5	1	1	1	5	5	5	5	2	2	2	2	2
-			4	4	4	8	8	8	2	2	2	2	3	3	3	5	5	5	0
0	1	2	3	4	4	5	6	7	8	9	10	10	11	12	13	14	15	15	**16**

b. **FIFO**

	9	5	4	2	5	8	9	1	2	4	5	5	3	8	2	5	3	2	0
-	9	9	9	2	2	2	2	1	1	1	5	5	5	5	2	2	2	2	0
-		5	5	5	5	8	8	8	2	2	2	2	3	3	3	5	5	5	5
-			4	4	4	4	9	9	9	4	4	4	4	8	8	8	3	3	3
0	1	2	3	4	5	5	6	7	8	9	10	10	11	12	13	14	15	15	**16**

5. **Given memory partitions of 100 KB, 500 KB, 200 KB, 300 KB and 600 KB (in order), how would each of the first-fit, best-fit and worst-fit algorithms place processes of 212 KB, 417 KB, 112 KB and 426 KB (in that order)? Which algorithm makes the most efficient use of memory?**

First-fit:

212K is put in 500K partition
417K is put in 600K partition
112K is put in 288K partition (new partition 288K = 500K - 212K)

426K must wait

Best-fit:

212K is put in 300K partition
417K is put in 500K partition
112K is put in 200K partition
426K is put in 600K partition

Worst-fit:

212K is put in 600K partition
417K is put in 500K partition
112K is put in 388K partition

426K must wait

In this example, best-fit turns out to be the best.