

**Problem Set (3): Memory Management**

---

**Study Questions**

[Q1] Consider a system that assigns memory partitions of size  $2^k$  with  $L \leq k \leq U$ . A list is maintained for free partitions of each size. For a request for memory block of size  $s$ , system assigns the first partition of size  $2^n$ , where  $2^{n-1} < s \leq 2^n$ . If necessary, system splits larger partitions in halves. When a partition becomes free it can be merged with an adjacent partition of the same size. What are the advantages and disadvantages of this technique?

[Q2] If an instruction takes  $1 \mu s$  and a page fault takes an additional  $n \mu s$ , give a formula for the effective instruction time if a page faults occurs every  $k$  instructions. Discuss the factors on which  $n$  and  $k$  will depend.

[Q3] Assuming a page size of 4 Kbytes and that a page table entry takes 4 bytes, how many levels of page tables would be required to map a 64-bit address space, if a table should fit into a single page?

[Q4] Discuss the difference between the following statements if virtual memory with paging is used:

```
for (i=0; i<n; i++)  
  for (j=0; j<n; j++)  
    A[i][j]=0;
```

```
for (j=0; j<n; j++)  
  for (i=0; i<n; i++)  
    A[i][j]=0;
```

[Q5] Consider the following replacement algorithm in a paged virtual memory system. A *reference bit* is associated with each page in the page table. Whenever a page is accessed, this reference bit is set to 1. When a page fault occurs, the reference bits of *all* the pages in physical memory are reset to 0. When a page is to be selected for replacement, the algorithm scans all reference bits in a predetermined order, and the first page found with a zero reference bit is the one replaced. If all reference bits are 1, then the page with the lowest physical address is replaced. Compare this algorithm with LRU with regard to cost of implementation and effectiveness.

---

**Problems:**

[P1] In a multi-programmed system, the memory areas were assigned as follows (with order starting from lower addresses):

- 24K resident part of the operating system.
- 200K assigned to process a.
- 50K assigned to process b.

100K assigned to process c.  
100K assigned to process d.  
120K assigned to process e.  
230K assigned to process f.  
200K free.

Then, the following sequence of events took place:

- Processes a, c and e terminated.
- Process g started requiring 80K of memory.
- Process h started requiring 100K of memory.
- Process i started requiring 150K of memory.

Which areas will the system allocate to the new processes in the case of best-fit, first-fit and next-fit algorithms?

---

**[P2]** A multi-programmed system runs the processes shown in the following list:

<u>Process</u>	<u>arrival time</u>	<u>process time</u>	<u>memory required</u>
A	0	6	4M
B	2	4	4M
C	3	5	3M
D	5	3	3M

Before the arrival of these processes, available free memory consisted of two areas of sizes 8M (at a lower address) and 6M. Find the average waiting time for the above processes assuming:

- SPN scheduling.
  - RR scheduling with  $q=1$ , and first-fit memory assignment.
  - RR scheduling with  $q=1$ , and best-fit memory assignment.
- 

**[P3]** Below is the listing of a short assembly language program for a computer with 512 bytes pages. The program is located at address 1020, and its stack pointer is at 8192 (the stack grows toward zero). Give the page reference string generated by this program. Each instruction occupies 4 bytes (i.e. 1 word), and both instruction and data references count in the reference string.

Load word 6144 into register 0  
Push register 0 into the stack  
Call a procedure at 5120, stacking the return address  
Subtract the immediate constant 16 from the stack pointer  
Compare register 1 to the immediate constant 4  
Jump if equal to 5152

---

**[P4]** Consider the page reference string: 1,2,3,4,2,1,5,6,2,1,2,3,7,6,3,2,1,2,3,6.

How many page faults would occur for FIFO and LRU replacement algorithms, assuming 4, 5 and 6 frames? (Assume all the frames initially empty.)

In each case, how will the virtual addresses in page 6 be modified into physical addresses?

---

**[P5]** A computer system uses virtual memory with paging with 24-bit virtual addresses and a page size of 4K bytes. A process accesses the following virtual addresses (given in hexadecimal):

002B2A - 006123 - 0037FF - 006127 - 00239C - 005000 - 002A1B - 00612B

a) Find the number of page faults in accessing the above addresses if 3 RAM frames are available for this process and FIFO replacement is used.

b) Repeat part (a) for LRU replacement.

c) Assume that the time to access a RAM location is  $t_m$ , and it takes  $t_f$  to respond to a page fault and load a page in RAM, with additional  $t_w$  to write a page to disk. Assume further that the contents of 4<sup>th</sup> and 5<sup>th</sup> addresses above were modified. Find the time required to execute the memory accesses in parts (a) and (b).

---

**[P6]** A computer system uses virtual memory with paging and uses the LRU replacement algorithm. It runs two processes A and B, and each of the two processes was assigned 3 physical memory frames. Process A accesses the virtual memory pages in the order 0-1-2-3-0-2-9, while process B accesses the pages 2-4-2-3-1-2.

Calculate the number of page faults when running the above two processes. Can this be improved by a different assignment of the 6 frames?

---