# CS490 Windows Internals

# Assignment 4 Solution

## Due: Oct 25. 2013

1. Investigate the page-replacement algorithms FIFO and LRU. Let us assume a system with four physical page frames that are initially free. Develop a diagram illustrating for both algorithms the flow of physical memory usage when pages are accessed in the order given below. Mark the occurrence of page faults within the diagram.

9 1 10 9 6 3 14 9 2 1 4 6 8 2 13 13 14 13 4 0 13 1 1 6 4

FIFO:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 1 | 10 | 9 | 6 | 3 | 14 | 9 | 2 | 1 | 4 | 6 | 8 | 2 | 13 | 13 | 14 | 13 | 4 | 0 | 13 | 1 | 1 | 6 | 4 |
| 9 | 9 | 9 | 9 | 9 | 3 | 3 | 3 | 3 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 1 | 1 | 1 | 1 | 1 | 14 | 14 | 14 | 14 | 4 | 4 | 4 | 4 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 1 | 1 | 1 | 1 |
|  |  | 10 | 10 | 10 | 10 | 10 | 9 | 9 | 9 | 9 | 6 | 6 | 6 | 6 | 6 | 14 | 14 | 14 | 14 | 14 | 14 | 14 | 6 | 6 |
|  |  |  |  | 6 | 6 | 6 | 6 | 2 | 2 | 2 | 2 | 8 | 8 | 8 | 8 | 8 | 8 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| F | F | F |  | F | F | F | F | F | F | F | F | F | F | F |  | F |  | F | F |  | F |  | F |  |

LRU:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 1 | 10 | 9 | 6 | 3 | 14 | 9 | 2 | 1 | 4 | 6 | 8 | 2 | 13 | 13 | 14 | 13 | 4 | 0 | 13 | 1 | 1 | 6 | 4 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 6 | 6 | 6 | 6 | 6 | 14 | 14 | 14 | 14 | 14 | 1 | 1 | 1 | 1 |
|  | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 4 |
|  |  | 10 | 10 | 10 | 10 | 14 | 14 | 14 | 14 | 4 | 4 | 4 | 4 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 |
|  |  |  |  | 6 | 6 | 6 | 6 | 2 | 2 | 2 | 2 | 8 | 8 | 8 | 8 | 8 | 8 | 4 | 4 | 4 | 4 | 4 | 6 | 6 |
| F | F | F |  | F | F | F |  | F | F | F | F | F | F | F |  | F |  | F | F |  | F |  | F | F |

1. Given is a 32-bit computer. Let the size of a page frame be 1kByte. The OS supports a maximum of 4Gbyte physical memory. The virtual memory management uses a single level of pages tables (i.e.; each process has a single page table that maps the entire virtual address space onto the physical address space). Let us further assume that page table cannot be paged out.

Our computer has a total of 64Mbyte of physical memory. The OS uses at least 3Mbyte memory that cannot be paged out. Answer the following questions:

* 1. How many entries does a page table have?

In order to address 4Gbyte of memory, and the page frame size is 1Kbyte, each page table has to have 222 entries (roughly 4M).

* 1. How much physical memory does a page table use? (Each page table entry shall contain the page frame number and 2 control bits.)

Since our system may support up to 4Gbyte of RAM, we need 22 bits to address all page frames. A page table entry is 24bit; the whole page table occupies 12Mbyte of memory.

* 1. If there are two processes running, what is the maximum size of the working set for one of these processes?

The page table for each process occupies 12Mbyte of memory; the OS uses at least 3Mbyte. Maximum working set size for one process is therefore 37Mbyte.

1. Suppose the page size in a computing environment is 4Kbyte. Give the page number and the offset for the following:
   1. 513 (a decimal number)

PN = 0; O = 513 (decimal)

* 1. 234567 (a decimal number)

PN = 57; O = 1095 (decimal)

* 1. 0xB457 (a hexadecimal number)

PN = 11; O = 1111 (decimal)

* 1. 0xCDEF (a hexadecimal number)

PN = 12; O = 3567 (decimal)

1. Explain the role of **standby**, **modified**, **free**, and **zero** page lists in the Windows memory manager.

The Windows memory manager uses standby, modified, free, and zero page lists to maintain information about pages in main memory. The following statements describe the characteristics of a page contained in the corresponding lists:

**Standby List**: The page previously belonged to a working set but was removed. The page wasn’t modified since it was last written to disk. The corresponding page table entry still refers to the physical page but is marked invalid and in transition.

**Modified List**: The page previously belonged to a working set but was removed. However, the page was modified while it was in use and its current contents haven’t yet been written to disk. The PTE still refers to the physical page but is marked invalid and in transition. It must be written to disk before the physical page can be reused.

**Free List**: The page is free but has unspecified dirty data in it. (These pages can’t be given as a user page to a user process without being initialized with zeros, for security reasons.)

**Zero List**: The page is free and has been initialized with zeros by the zero page thread.