Fall 2015 COMP 3511 Operating Systems

Lab #8

Outline

- Thrashing
- Working-Set model
- File system
- File system implementation

Q. 1

- Please state the disadvantages and advantages with the following page replacement policies over page fault rate, belady’s anomaly, easy to implement:
  a) FIFO
  b) LRU
  c) Optimal

Q. 1 Disadvantages

- FIFO
  - It suffers from Belady’s anomaly which states that for some page-replacement algorithms, the page-fault rate may ***increase as the number of allocated frames*** increases.
  - Page fault rate is high in average.
- LRU
  - Not easy to implement LRU algorithm.
  - Requires substantial hardware assistance, e.g. TLB registers.
  - Problem in determining the frame order by the time of last use.
- Optimal
  - Needs future knowledge of the reference string.
  - It is the very difficult or near impossible to implement.

Q. 1 Advantages

- FIFO
  - Most simple policy for page replacement algorithm, easy to implement.
- LRU
  - LRU policy is often used as a page replacement algorithm and is a good one with less page fault. It's performance in regards to page fault rate is between FIFO and Optimal.
  - It does not suffer from Belady's anomaly.
  - It is easy to choose the pages that have already page faulted and not in use for a long period. So it is feasible to implement.
- Optimal
  - Has the lowest page fault rate.
  - It never suffers from Belady's anomaly.
  - Can be used for comparison studies (even a new algorithm may not be optimal, on average it is within 4.7% of optimal).

Q. 2

- Provide a short definition for each of the following terms:
    - Page fault
    - Working Set

Q. 2

- Page fault

  A page fault occurs when the CPU tries to use a logical address for which a corresponding physical address does not exist, that is, the process page that contains the logical address had not been loaded into physical memory. The OS must load the logical page into physical memory to allow the process to continue its execution.

- Working Set

  The set of pages required by a process to execute, that is, according to locality of reference, the set of pages required by the process in memory to keep the number of page faults to a minimum.

Q. 3

- How does a system detect thrashing? Once it detect thrashing, what can the system do to eliminate this problem?

Q. 3

- If a process does not have enough pages, thrashing occurs as a high paging activity due to the high page fault rate, hence rapidly exchanging data in memory for data on disk. The continuously page fault may be more serious when OS observed a lower CPU utilization by introducing more new processes to the system. The system can detect thrashing by evaluating the level of CPU utilization as compared to the level of multiprogramming.

- System does following things to eliminate this problem:
    1) It can be eliminated by reducing the level of multiprogramming.
    2) Thrashing can also be removed by using local replacement algorithm.

Q. 4

- Consider a demand-paged computer system where the degree of multiprogramming is currently fixed at four processes. The system was recently measured to determine utilization of CPU and the **paging** disk (which means that the regular I/O to the disk is not included in the statistics). The results are one of the following alternatives. For each case, what is happening? Can the degree of multiprogramming be increased to increase the CPU utilization?
  a) CPU utilization 13 percent; disk utilization 97 percent
  b) CPU utilization 87 percent; disk utilization 3 percent
  c) CPU utilization 13 percent; disk utilization 3 percent

Q. 4

a) Thrashing is occurring, since the disk utilization is very high while the CPU is underutilized. Processes are spending most of their time paging. The degree of multiprogramming cannot be increased. On the contrary, one or more processes should be suspended to allow increase in the utilization of the CPU.

b) CPU utilization is sufficiently high to leave things alone. Any increase in the degree of multiprogramming could lead to thrashing.

c) Increase the degree of multiprogramming. The CPU is available for executing additional processes.

Working-Set Model

- The locality model states that, as a process executes, it moves from locality to locality. A locality is defined as a set of pages that are actively used together.

- Working-Set model is based on the locality
  - $\Delta \equiv$ working-set window $\equiv$ a fixed number of page references
  - WSS_i (working set of Process P_i) = total number of pages referenced in the most recent Δ (varies in time)
  - $D = \Sigma\, WSS_i \equiv$ total demand for frames (by all processes)

Working-Set Model

- Consider the parameter Δ used to define the working-set window in the working-set model.
    - What is the effect of setting Δ to a small value on the page fault frequency and the number of active (non-suspended) processes currently executing in the system?
    - What is the effect when Δ is set to a very high value?

Working-Set Model

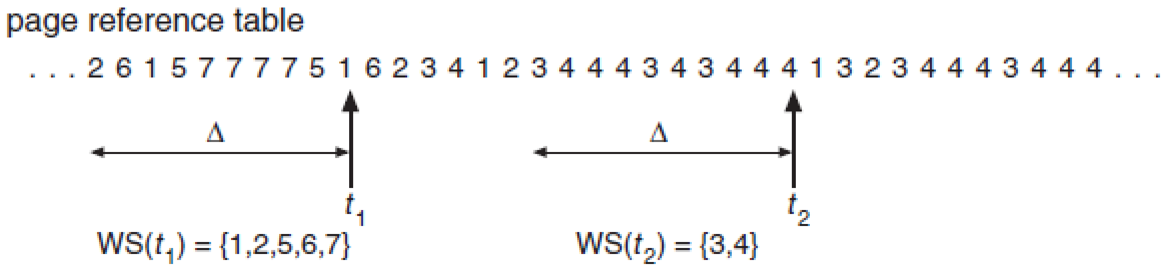
- When Δ is set to a small value
  - the set of resident pages for a process might be underestimated
  - allowing a process to be scheduled even though all of its required pages are not resident
  - This could result in a large number of page faults

Working-Set Model

- When Δ is set to a large value
  - a process’s resident set is overestimated
  - this might prevent many processes from being scheduled even though their required pages are resident
  - However, once a process is scheduled, it is unlikely to generate page faults since its resident set has been overestimated

Working-Set Window

- A working set is the set of pages in the most recent working-set window Δ are actively referencing***.***
- The working-set window is a moving window. At each memory reference, a new reference appears at one end, and the oldest reference drops off the other end.
- A page is in the working set if it is referenced anywhere in the working-set window.
- *For example, in the following working-set model:*



- if Δ = *10 memory references, then the working set at time t1 is {1,2,5,6,7}*. By time *t2, the working set has changed to {3,4}*

Q. 5

- Given a process that references virtual pages during its execution as indicated in the table below and an observation window size Δ of 5; complete the table below to show the process working set changes during its execution.

Page Reference	Working set
3	3
2	3,2
4	3,2,4
3	3,2,4
4	3,2,4
2	3,2,4
2	3,2,4
3	3,2,4
4	
5	
6	
7	
6	
7	5,6,7

Q. 5

- Given a process that references virtual pages during its execution as indicated in the table below and an observation window size Δ of 5; complete the table below to show the process working set changes during its execution.

Page Reference	Working set
3	3
2	3,2
4	3,2,4
3	3,2,4
4	3,2,4
2	3,2,4
2	3,2,4
3	3,2,4
4	3,2,4
5	3,2,4,5
6	3,2,4,5,6
7	3,4,5,6,7
6	4,5,6,7
7	5,6,7

Q. 6

- Can a page be in two working sets at the same time? Please explain.

Q. 6

- If pages can be shared, it is possible that a page is in two working sets at the same time.
- For example, if two users of a timesharing system are running the same editor at the same time, and the program text is shared rather than copied, some of those pages may be in each user's working set at the same time.

FILE SYSTEM

Q. 7

- What is the difference between a mandatory lock and an advisory file lock?

Q. 7

- What is the difference between a mandatory lock and an advisory file lock?
  - Mandatory locking requires that the operating system does not allow access to any file that is locked, until it is released, even if the program does not explicitly ask for a lock on the file. An advisory file locking scheme will not prevent access to a locked file, and it is up to the programmer to ensure that locks are appropriately acquired and released.

Q. 8

- Specify the primary advantages the acyclic graph directory has compared with the tree directory.
- Several aspects need to be considered when using acyclic-graph directory, please state two of them.

Q. 8

- Specify the primary advantages the acyclic graph directory has compared with the tree directory.
  - Acyclic graph directory permits the sharing of files or directories. And it is more flexible than the simple tree structure.
  - The relative simplicity of the algorithms to traverse the graph and to determine if a file can be deleted when there are no more references to a file.
- Several aspects need to be considered carefully when using acyclic-graph directory, please state two of them.
  - A file may have multiple absolute path names hence can be accessed by different file names.
  - When can the space allocated to a shared file be deallocated and reallocated to other files.

Q. 9

- Consider the listing of command “ls –liaR” that provides the contents of a Linux file system on the next page. The arguments of ls are described as below:

  -l: long listing format

  -i: provide the inode number

  -a: list all entries in directories including the hidden ones

  -R: recurs into subdirectories

Q. 9

```
[root@sitedev mnt1]# ls -liaR .
total 7
      1 drwxr-xr-x    4 root     root         128 Mar 27 09:24 .
  96770 drwxr-x---    8 root     root        4096 Mar 27 09:47 ..
      2 drwxr-xr-x    5 test1    244          160 Mar 27 09:24 dir1
      3 drwxr-xr-x    4 test1    244          160 Mar 27 09:30 dir2

./dir1:
total 5
      2 drwxr-xr-x    5 test1    244          160 Mar 27 09:24 .
      1 drwxr-xr-x    4 root     root         128 Mar 27 09:24 ..
      4 drwxr-xr-x    2 test1    244           96 Mar 27 09:36 ad
      5 drwxr-xr-x    2 test1    244           96 Mar 27 09:29 bd
      6 drwxr-xr-x    2 test1    244           96 Mar 27 09:30 cd
```

Q. 10

- What are access permissions for ‘dir2’? What’s its user id, group id, link count, size in bytes, and time modified?

Q. 10

- What are the access permissions for ‘dir2’? What’s its user id, group id, link count, size in bytes, and time modified?
  - rwxr-xr-x: owner has permissions of read, write, execute
    group has permissions of read and execute
    others have permissions of read and execute
  - Its user id is test1; group id is 244; number of link count is 4 which means the number of times it is referenced by the directory or in this case the number of subdirectories (including . and ..); size is 160 bytes; the time modified is Mar 27 09:24.

Q. 11

- What are three common methods for remote file-sharing?

Q. 11

- The first implemented method involves manually transferring files between machines via programs like ftp. The second major method uses a distributed file system (DFS), in which remote directories are visible from a local machine. In the third method, a browser is needed to access remote files on the World Wide Web, and separate operations (essentially a wrapper for ftp) are used to transfer files. The DFS method involves a much tighter integration between the machine that is accessing the remote files and the machine providing the files.

Q. 12

- List some examples of sequentially accessed files and direct-access files.

Q. 12

- List some examples of sequentially accessed files and direct-access files
  - Sequential: word processors, music players, video players, and web servers
  - Random-access: databases, video and audio editors

FILE SYSTEM IMPLEMENTATION

Q. 13

- Consider a file system that uses inodes to represent files. Disk blocks are 8-KB in size and a pointer to a disk block requires 4 bytes. This file system has 12 direct disk blocks, plus single, double, and triple indirect disk blocks. What is the maximum size of a file that can be stored in this file system?

Q. 13

- Consider a file system that uses inodes to represent files. Disk blocks are 8-KB in size and a pointer to a disk block requires 4 bytes. This file system has 12 direct disk blocks, plus single, double, and triple indirect disk blocks. What is the maximum size of a file that can be stored in this file system?

  - (12 * 8 /KB/) + (2048 * 8 /KB) + (2048 * 2048 * 8 /KB/) + (2048 * 2048 * 2048 * 8 /KB) = 64 terabytes

Q. 14

- What are the advantages of the variation of linked allocation that uses a FAT to chain together the blocks of a file?

Q. 14

- What are the advantages of the variation of linked allocation that uses a FAT to chain together the blocks of a file?
  - The advantage is that while accessing a block that is stored at the middle of a file, its location can be determined by chasing the pointers stored in the FAT as opposed to accessing all of the individual blocks of the file in a sequential manner to find the pointer to the target block. Typically, most of the FAT can be cached in memory and therefore the pointers can be determined with just memory accesses instead of having to access the disk blocks.

Q. 15

- Contrast the performance of the three techniques for allocating disk blocks (contiguous, linked, and indexed) for both sequential and random file access

Q. 15

- Contrast the performance of the three techniques for allocating disk blocks (contiguous, linked, and indexed) for both sequential and random file access:
  - Contiguous Sequential - Works very well as the file is stored contiguously.
  - Sequential access - Simply involves traversing the contiguous disk blocks.
  - Contiguous Random -Works very well as you can easily determine the adjacent disk block containing the position you wish to seek to.
  - Linked Sequential - Satisfactory as you are simply following the links from one block to the next.

Q. 15 Cont.

- Linked Random - Poor as it may require following the links to several disk blocks until you arrive at the intended seek point of the file.
- Indexed Sequential - Works well as sequential access simply involves sequentially accessing each index.
- Indexed Random - Works well as it is easy to determine the index associated with the disk block containing the position you wish to seek to