ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Operating Systems and Networks	Spring 2016
Assignment 4	Due 29 March 2015

1. **Paging**
   (a) **Paging Problem I**

      i. Consider the following two-dimensional array:
      `int X[32][32];`
      Suppose that a system can accomodate 4 pages in main memory and each page is 512 bytes long (assume an integer variable is of size 8 bytes). Suppose The `X` array is stored in row-major order (i.e., `X[0][1]` follows `X[0][0]` in memory). Which of the following two code fragments will generate the less number of page faults? Explain and compute the total number of page faults for each code.
      Fragment 1:
      ```
      for (int j=0; j<32; j++)
        for (int i=0; i<32; i++)
         X[i][j]++;
      ```
      Fragment 2:
      ```
      for (int i=0; i<32; i++)
        for (int j=0; j<32; j++)
         X[i][j]++;
      ```

   (b) **Paging Problem II**

      Suppose that a virtual page reference stream contains repeatative long sequences of page references and then occasionally followed by a page reference outside the sequence. For example, the follwoing code:
      ```
      for (int i=0; i<n; i++){
        for (int j=0; j<S; j++)
           A[j*PAGE_SIZE]++;
        B[random_integer]++;
      }
      ```
      i. What should be the lower bound of the number of pages the main memory allows for this program so that page replacement algorithms like LRU and FIFO does not become ineffective (every access in the loop nest results in fault)?
      ii. If this program were allocated F page frames where $1 < F < S + 1$, describe a page replacement approach that would perform much better than the LRU, FIFO algoirthm.

   (c) **Thrashing Problem I**

      i. If there are N processes running in parallel in a processor, we call N "the degree of multiprogramming". What do you expect the CPU utilization will be if we keep increasing the number N. Can these N processes running in parallel cause thrashing? Explain both of your answers.
      ii. How can an OS detect thrashing? What action can the OS take to avoid thrashing?

   (d) **Thrashing Problem II**
      Consider a demand-paging system with the following fractions of time consumptions for running N processes (asssume the phenomenons are non-overlapping):

      CPU utilization = 20%
      Paging activity (including load from disk) = 75%
      Other I/O devices =5%

For each of the following, say whether it will (or is likely to) improve CPU utilization. Explain your answers.

- install a faster CPU
- increase the degree of multiprogramming
- decrease the degree of multiprogramming
- install more memory
- install a faster hard disk

2. **File System**

   i. Some OSes provide a system call "rename" to give a file a new name. Is there any difference at all between using this call to rename a file and just copying the file to a new file with the new name, followed by deleting the old one?

   ii. What are the differences between hard links and symbolic links?