COMP 530 Introduction to Operating Systems

Fall 2017  
Kevin Jeffay

Worksheet 17, November 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Your Name: |  | You worked with: |  | +1/blank/-1: |  |
|  | Aaron Zhang |  | John Espenhahn |  | +1 |  |
|  |  |  | Nils Persson |  | +1 |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

1. A famous processor/computer system was designed that did not have a reference bit in the page tables it supported. Nonetheless, developers wanted to run operating systems on this system whose virtual memory management system required a reference bit. Develop a scheme the operating system can use to simulate the effect of having a reference bit. Describe how a clock page replacement algorithm would work with your simulated reference bit. Discuss what the cost of your simulation would be in terms of system overhead (compared to the operation of the virtual memory system if the hardware had a reference bit).

Use demand paging and simulate the effect of the reference bit in software. When a page is loaded in memory, record this fact in software and set the page’s residence bit to invalid.

When the page is accesses a page fault now occurs. Record the fact that the page has been accesses and set the resident bit to valid.

In the clock algorithm, for each resident page, clear the resident bit if set… otherwise replace the current page.

Cost of this simulation -> one extra system call per each page of the VM used per complete iteration of the clock algorithim

Effectively an extra system call per page fault per resident page in the worst case

2. Consider a virtual memory system wherein it is possible for a page table in process *P* to have one or more valid entries, each of which contains the same frame number as some valid entry in a page table for process *Q.* For example, in the simplest case, the *ith* row of *P*’s page table is valid and contains the same frame number that is found in the *jth* row of *Q*’s page table (which is also a valid entry).

*a*) What is the effect of allowing valid entries in two processes page tables to contain the same page frame number? What would the effect of updating some byte in page *i* be on page *j*?

*b*) Explain how an operating system can allow this behavior to occur as a means to improve the performance of some operating system function. (For example, to improve the performance of a function that we’ve seen in lecture such as *dispatch*, *context\_switch*, *up*, *down*, *fork*, *exec*, *join*, *send*, *receive*,…) You need to be specific in your answer.

A)

Parameter passing, same technique used to pass data between two processes. Copy is made when the write is done

b)

code should never change, so set it the same frame in physical memory