Homework 9

9.3
A certain computer provides its users with a virtual-memory space of $2^{32}$ bytes. The computer has 218 bytes of physical memory. The virtual memory is implemented by paging, and the page size is 4096 bytes. A user process generates the virtual address 11123456. Explain how the system establishes the corresponding physical location. Distinguish between software and hardware operations.

9.5
Discuss situations under which the least frequently used page-replacement algorithm generates fewer page faults than the least recently used page-replacement algorithm. Also discuss under what circumstance does the opposite holds.

9.6
Discuss situations under which the most frequently used page-replacement algorithm generates fewer page faults than the least recently used page-replacement algorithm. Also discuss under what circumstance does the opposite holds.

9.7
Consider a demand-paging system with the following time-measured utilizations:

<table>
<thead>
<tr>
<th></th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU utilization</td>
<td>20%</td>
</tr>
<tr>
<td>Paging disk</td>
<td>97.7%</td>
</tr>
<tr>
<td>Other I/O devices</td>
<td>5%</td>
</tr>
</tbody>
</table>

Which (if any) of the following will (probably) improve CPU utilization? Explain your answer.

a. Install a faster CPU.
b. Install a bigger paging disk.
c. Increase the degree of multiprogramming.
d. Decrease the degree of multiprogramming.
e. Install more main memory.
f. Install a faster hard disk or multiple controllers with multiple hard disks.
g. Add prepaging to the page fetch algorithms.
h. Increase the page size.

9.9
A page-replacement algorithm should minimize the number of page faults. We can do this minimization by distributing heavily used pages evenly over all of memory, rather than having them compete for a small number of page frames. We can associate with each page frame a counter of the
number of pages that are associated with that frame. Then, to replace a page, we search for the page frame with the smallest counter.

a. Define a page-replacement algorithm using this basic idea. Specifically address the problems of
(1) what the initial value of the counters is,
(2) when counters are increased,
(3) when counters are decreased, and
(4) how the page to be replaced is selected.

b. How many page faults occur for your algorithm for the following reference string, for four page frames?

1, 2, 3, 4, 5, 3, 4, 1, 6, 7, 8, 7, 8, 9, 7, 8, 9, 5, 4, 5, 4, 2.

c. What is the minimum number of page faults for an optimal page-replacement strategy for the reference string in part b with four page frames?

9.12
Consider the parameter \( \Delta \) used to define the working-set window in the working-set model. What is the effect of setting \( \Delta \) 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 \( \Delta \) is set to a very high value?

9.13
The slab allocation algorithm uses a separate cache for each different object type. Assuming there is one cache per object type, explain why this doesn’t scale well with multiple CPUs. What could be done to address this scalability issue?

9.15
The Catalan numbers are an integer sequence \( C_n \) that appear in tree enumeration problems. The first Catalan numbers for \( n = 1, 2, 3, \ldots \) are 1, 2, 5, 14, 42, 132, \ldots A formula generating \( C_n \) is

\[
C_n = \frac{1}{(n+1)} \binom{2n}{n} = \frac{(2n)!}{(n+1)!n!}
\]

Design two programs that communicate with shared memory using the Win32 API as outlined in Section 9.7.2. The producer process will generate the Catalan sequence and write it to a shared memory object. The consumer process will then read and output the sequence from shared memory.

In this instance, the producer process will be passed an integer parameter on the command line specifying the number of Catalan numbers to produce, i.e., providing 5 on the command line means the producer process will generate the first 5 Catalan numbers.