Homework 6 - 5/21/2019

Friday, May 17, 2019 4:21 PM

Daniel Yan

1) 5.32 (you can use the Java wait-notify constructs to solve this problem)
2) 8.11
3) 8.21
4) 8.23

Exercise 5.32
A file is to be shared among different processes, each of which has a unique number. The file can be accessed simultaneously by several processes, subject to the following constraint: the sum of all unique numbers associated with all the processes currently accessing the file must be less than n. Write a monitor to coordinate access to the file.

```
import java.io.*;
import java.util.*;
public class FileShare {
 int total, nSum;

 FileShare(int n)
 {
  this.nSum = n;
 }

 public synchronized void access(int i)
 {
      if(i + total >= nSum)
      {
       try{
          wait();
        }
       catch (InterruptedException e) {}
      }
      total += i;
 }
 public synchronized void release(int i){
      total -= i;
      notify();
 }
 public static void main(String[] args){
 }
}
```

Exercise 8.11
Given six memory partitions of 300KB, 600KB, 350KB, 200KB, 750KB,and 125KB(in order), how would the first-fit, best-fit, and worst-fit algorithms place processes of size 115KB, 500KB, 358KB, 200KB ,and 375KB (in order)? Rank the algorithms in terms of how efficiently they use memory.

Ranking of Memory Efficiency:
Both First & Best Fit have same memory allocation, First Fit should have slightly better performance due to not crawling through each memory partition.

1. **First Fit:**
   P1: 115 KB -> 300 KB
   P2: 500 KB -> 600 KB
   P3: 358 KB -> 750 KB
   P4: 200 KB -> 350 KB
   P5: 375 KB -> not able to allocate
2. **Best Fit:**
   P1: 115 KB -> 125 KB
   P2: 500 KB -> 600 KB
   P3: 358 KB -> 750 KB
   P4: 200 KB -> 200 KB
   P5: 375 KB -> not able to allocate
3. **Worst Fit:**
   P1: 115 KB -> 750 KB
   P2: 500 KB -> 600 KB
   P3: 358 KB -> not able to allocate
   P4: 200 KB -> 350 KB
   P5: 375 KB -> not able to allocate

Exercise 8.21

The BTV operating system has a 21-bit virtual address, yet on certain embedded devices, it has only a 16-bit physical address. It also has a 2-KB page size. How many entries are there in each of the following?

A. A conventional, single-level page table
   - 2KB page size = 2^{11}, or 11 bits for page size. Using a 21 bit virtual address or 2^{21} for 2MB. Subtract $\frac{2^{21}}{2^{11}} = 2^{10}$ to ge the remainder of 2^{10} or 1024 entries in the page table .

B. An inverted page table
   - Using the physical address, $\frac{2^{16}}{2^{11}} = 2^5$ or 32 entries.

Exercise 8.23

Consider a logical address space of 256 pages with a 4-KBpage size, mapped onto a physical memory of 64 frames.

A. How many bits are required in the logical address?
   - 256 pages = 2^8 or 8 page bits, with a 4KB page size = 2^{12}, thus we require $2^{12} * 2^8 = 2^{20}$ addresses or 20 bits for the logical address.

B. How many bits are required in the physical address?
   - 64 frames = 2^6 or 6 frame bits, with a 4KB page size = 2^{12}, thus we require $2^{12} * 2^6 = 2^{18}$ addresses or 18 bits for the physical address.