Theory – Assignment5

4061

Cole Wallin

Problem 1: (5 points) Consider a system with three processes, named P, Q, and R. Each process continuously prints its color. We now require that the printing by the processes be coordinated such that the output produced would be P:RED Q:WHITE R:BLUE P:RED Q:WHITE R:BLUE P:RED Q:WHITE R:BLUE … Using semaphores, synchronize these three process such that processes P, Q, and R execute their print statements in the above order. Insert appropriate synchronization code before and after the each print statement:

```
Int turn = 0; // Will change between 0, 1, and 2

Mutex_t mutex;

Process P {
        while ( true ) {
                while (turn != 0); //Wait
                lock(mutex);
                turn = (turn + 1) % 3;
                print("P:RED");
                unlock(mutex);
        } //end of process P
}

Process Q {
        while ( true ) {
                while (turn != 1); //Wait
                lock(mutex);
                turn = (turn + 1) % 3;
                print("Q:WHITE");
                unlock(mutex);
        } //end of process Q
}

Process R {
        while ( true ) {
                while (turn != 2); //Wait
                lock(mutex);
                turn = (turn + 1) % 3;
                print("R:BLUE");
                unlock(mutex);
        }
} //end of process R
```

Problem 2: (5 points) Suppose that a system has two processes, each of which needs 6 seconds of CPU time and 12 seconds for I/O. I/O operations are processed sequentially.

(a) How long will it take to complete both these processes if they run sequentially? – 36 seconds

(b) What are the best case and worst case completion times for these processes if they run concurrently, using multiprogramming? – Best case: 12 seconds

Worst Case: 24 seconds

Question 3: (5 points) Consider a system in which a page can store 200 integers. On this system a small program that operates on a two-dimensional matrix A is executed. The program code resides in page 0, which corresponds to addresses 0 through 199. This page is always kept in the physical memory. A is defined as: int A[ ] [ ] = int [10][200]; where A[0][0] is at the logical address 200 and the matrix is stored in the memory in the row-major form.

Consider the following two ways to initialize this matrix. For each of these two cases answer the following:

Identify the pattern in page reference strings these two initialization routines would generate for accessing matrix A. You may just identify the pattern that gets repeated because there may be several hundred page references.

Initialization 1:

```
for (int i =0; i< 10; i++)
    for (int k =0; k < 200; k++)
        A[ i ] [ k ] = 0;
```

Pattern in page numbers accessed: Access the first page 200 times, then the second page 200 times, then the third page 200 times, then the fourth page 200 times, then the fifth page 200 times, then the sixth page 200 times, then the seventh page 200 times, then the eighth page 200 times, then the ninth page 200 times, and then the tenth page 200 times.

Initialization 2:

```
for (int k =0; k < 200; k++)
        for (int i =0; i< 10; i++)
                A[ i ] [ k ] = 0;
```

Pattern in page numbers accessed: Sequentially access the pages 1 through 10 and repeat this process 200 times so it looks something like this: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, . . .

Problem 4 (5 points): Consider a Unix style file allocation on disk with total 13 storage pointer entries in the inode. In the inode, the first 10 storage pointers point directly to file data blocks, the other three are pointers to indirect pointer blocks, doubly-indirect pointer blocks, triple-indirect pointer blocks. Assume that the file data - block size is 1028 bytes, and an indirect block contains 256 file data-block addresses. What's the maximum file size which this system can store?

Max File Size = $(10*1028) + (256*1028) + (256^2*1028) + (256^3*1028) = 17,314,622,504$ bytes