**Tran Nguyen**

**CS 3343 Operating Systems Final Exam 100 points**

**Email your completed exam in Microsoft word format to me at** [**harringp@nsuok.edu**](mailto:harringp@nsuok.edu)

**One exam submission per student**

**Due May 6 No late exams accepted**

**13 Questions**

**Part 1: Answer the following questions:**

1. *(5 points)* What are the two parts of an address generated for paging to take place?

Each CPU-generated address is divided into two parts: the page number used as an index to the corresponding element in the page table. The relative address on the page combines with the page's starting address to create the physical address used by the memory manager. The size of the page is determined by the hardware. To make it easier to parse the virtual address into page numbers and relative addresses, the size of a page is usually a power of 2.

1. *(5 points)* List the top three highest windows thread priority levels.

Priorities have no effect on processor speed where higher priority process will not run faster or even use more CPU time. Threads, which are part of the processes, are what runs. A program can also change the underlying priority of each of its threads, within the scope determined by the priority of the process. But most programs do not mind. If you have more CPU cores, things get more interesting, and priorities are generally less efficient. Let’s say you have two threads you want to run, and let’s say you have two or more CPU cores doing nothing else with a priority equal to or higher than those threads. Your two threads will then receive 100% of the cores, regardless of their respective priority.

1. *(5 points)* Which level of priority do you believe *computer virus* threads or processes use to steal information?

Malware can be a piece of code that can often be in the form of software intentionally designed to affect a computer system. Once installed into a computer system, it can access the computer system's resources, can share data to some remote servers without user intervention or can monitor expenses. user details.

1. *(5 points)* What is the critical section problem for the windows operating system?

Critical section problem is a piece of code or the part of a program that can only be executed by a certain thread at a time, tries to access shared resources. If more than one thread is executing this code, an error occurs. We can say it is problem about the job find a way to design a protocol that for the process that can grating with each other to accomplish its task.

1. *(5 points)* What could a computer virus do to interfere in the critical section?

Computer virus is a piece of program code to infiltrate your computer for the purpose of stealing information, deleting data, ... Viruses can spread into your computer. computer via USB, phone, external hard drives. If these devices have been infected with viruses, the computer will also be infected with the virus without protection.

1. *(5 points)* What are the challenges of multi-core programming versus single-core programming?

The challenges of multi-core programming are: Dividing Activities, Balance, Data splitting, Data dependency, Testing and Debugging. The challenge is also to modify program to take advantage of multicore system.

1. *(5 points)* What are the two hardware components needed for contiguous memory allocation?

Main memory and Relocation registers

1. *(5 points)* What are kernel threads?

You can imagine Kernel as a translator. It converts input and output requests from software into an instruction set for the CPU and GPU. Simply put, it's a layer in between the software and the hardware that makes anything work. Management Kernel: CPU / GPU, Memory, Input / Output devices or IO. These are some Kernel thread: Monolithic Kernel, Micro Kernel, Hybrid Kernel, ...

1. *(5 points)* Which could a computer virus use to cause more damage to the operating system: dynamic loading or linking?

I think the virus can cause more damage to dynamic loading, because useful when large amounts of code are needed to handle infrequently occurring cases, there are a high percentage change it contains a few or many viruses in the code.

**Part 2: Java Programming:**

Given the java programs **RAF.java** and **MakeBACKING\_STORE.java** below and also included with this email:

1. *(15 points)* Change the RAF.java so that the user can input the *number* and *filename* instead of having to pass command-line arguments via the *args[ ]* .
2. *(10 points)* After running the above changes and observing the results, modify the RAF.java so that the random access file is listed as *write-only*. Rerun the program and describe the results. Paste the screenshot of your results below.

Text

Description automatically generated

So I guess at beginning the code still don’t have to pass command-line via the args, and still find the file and have user input file name as write only.

1. *(15 points)* What happens if you change the random access file so that it also has only the “r” added to it (e.g., as in Linux privilege-level access)?

I think in some case, you can have “r” instead of “rw” while r is only for read mode and rw is reading and writing mode.

1. *(15 points)* Then modify RAF.java so that it also contains the additional code to write the file for the BACKING\_STORE file. Are there any conflicts?

When I run the file with some change, on my screen the code still run normally but when I started type something, the code did not display anything and just go through “Process finished with exit code 0”

**Copies of the two java programs are below and also included with this email.**

import java.util.Scanner;  
  
public class RAF  
{  
 public static void main(String[] args) throws java.io.IOException {  
 Scanner sc = new Scanner(System.*in*);  
 String inputFileName;  
 inputFileName = sc.nextLine().trim();  
 File input = new File( inputFileName);  
 Scanner scan = new Scanner( input );  
  
 /\*if (args.length != 1) {  
 String userName = sc.nextLine();  
 System.err.println("Usage: java RAF <integer value>");  
 System.exit(0);  
 }\*/  
  
 // the file representing the simulated disk  
 File fileName;  
 RandomAccessFile disk = null;  
 int val;  
  
 int number = Integer.*parseInt*(args[0]);  
  
 try {  
 fileName = new File("RAFFile");  
 disk = new RandomAccessFile(fileName, "r");  
  
 */\*\*  
 \* now write an int to the file  
 \* beginning at position 100  
 \* NOTE - integer is 4 bytes  
 \*/* // seek to byte position 100  
 disk.seek(100);  
  
 // write a 4-byte integer  
 disk.writeInt(number);  
  
 // again seek to byte position 100  
 disk.seek(100);  
  
 */\*\*  
 \* read() returns a byte, but since bytes  
 \* in Java are signed, we use an integer  
 \* to store its value to obtain the signed  
 \* value of the byte.  
 \*/* for (int i = 0; i < 4; i++) {  
 val = disk.read();  
 System.*out*.println("byte " + i + ": " + val);  
 }  
  
 // seek again to byte position 100  
 disk.seek(100);  
  
 // now read 4 bytes  
 byte[] data = new byte[4];  
 disk.read(data);  
 for (int i = 0; i < 4; i++)  
 System.*out*.println("byte " + i + ": " + data[i]);  
 }  
 catch (IOException e) {  
 System.*err*.println ("Unable to start the disk");  
 System.*exit*(1);  
 }  
 finally {  
 disk.close();  
 }  
 }  
}

package com.company;  
  
import java.io.File;  
import java.io.IOException;  
import java.io.RandomAccessFile;  
import java.util.Scanner;  
  
public class MakeBACKING\_STORE {  
 public static void main(String[] args) throws java.io.IOException {  
 Scanner sc = new Scanner(System.*in*);  
 String inputFileName;  
 inputFileName = sc.nextLine().trim();  
 File input = new File( inputFileName);  
  
 // the file representing the simulated disk  
 File fileName;  
 RandomAccessFile disk = null;  
  
 try {  
 fileName = new File("BACKING\_STORE");  
 disk = new RandomAccessFile(fileName, "rw");  
  
 for (int i = 0; i < 256 \* 256 / 4; i++) {  
 disk.writeInt(i);  
 }  
 } catch (IOException e) {  
 System.*err*.println("Unable to create the file BACKING\_STORE");  
 System.*exit*(1);  
 } finally {  
 disk.close();  
 }  
 }  
}