Kristianstad University
SE-291 88 Kristianstad
+46 44 20 30 00
www.hkr.se

Exercise 1 - Operating Systems

In this exercise we will look more about processes, and also write some simple programs for creating processes and threads in Java. The exercise is divided into four parts. The first part gives some theoretical questions about the material covered in lecture 1-3. These questions are preferable worked out before the exercise session. The second part is a practical exercise about processes in Windows. Here we will work with the task manager and the different tools for process management. The third part is about processes and threads in Java, and finally the last part will focus on scheduling. In this part scheduling algorithms will be compared, using a couple of metrics.

1. One user can read the private data of another user - privacy.

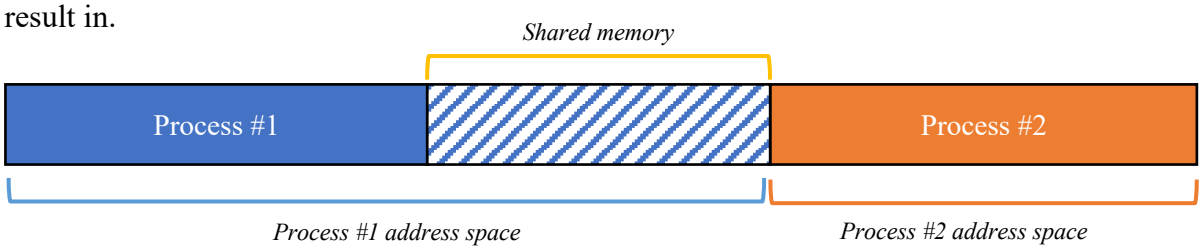
2. One user can corrupt the private data of another user - integrity.

3. One user can prevent another user from getting anything done - denail of service.

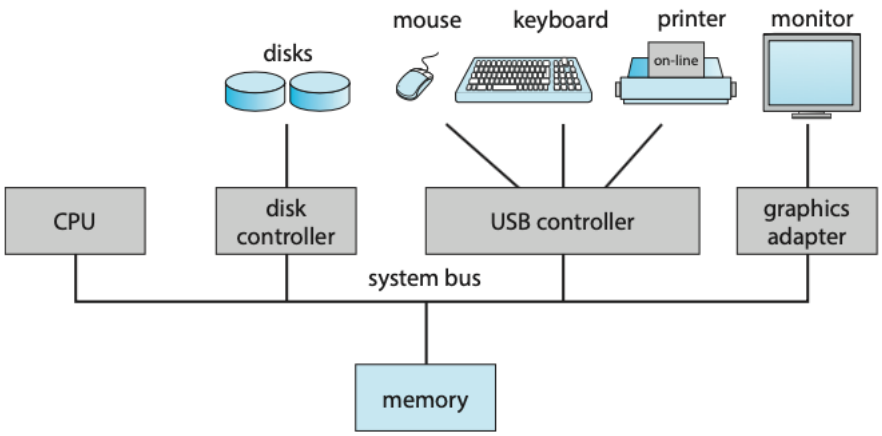
Part 1- Theoretical Questions

Answer the following questions.

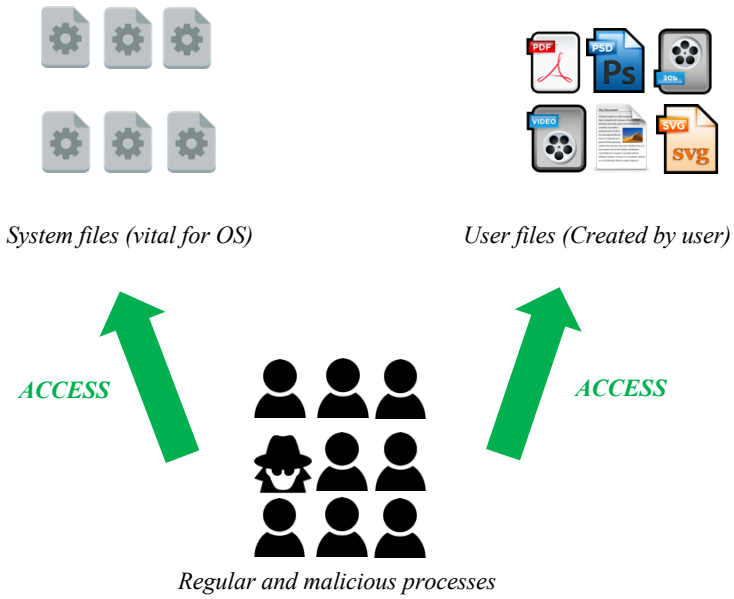
1. In a multiprogramming, several processes share the system simultaneously (*share the same memory block simultaneously*). Describe two security problems that this situation can result in.



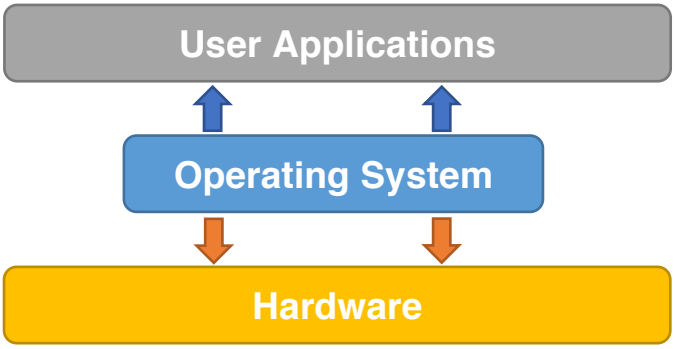
2. What is the purpose of interrupts and traps? (*Trap is a software generated interrupt*) Can traps be generated intentionally by a user program? If so, for what purpose?



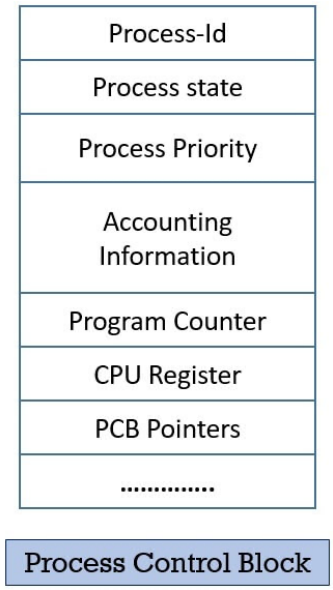
3. Is it possible to construct a secure operating system for a computer system that do not provide a privileged mode of operation? Give arguments both that it is and that it is not possible.



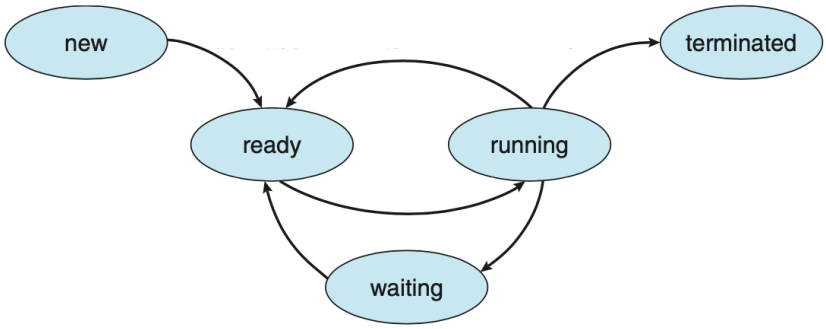
4. The computer system can roughly be divided into four components (*User, Application programs, Operating System, Hardware).* Give an example of user interface, tell the main purpose of operating system, give examples of application programs, name the purpose of device driver and device controller.



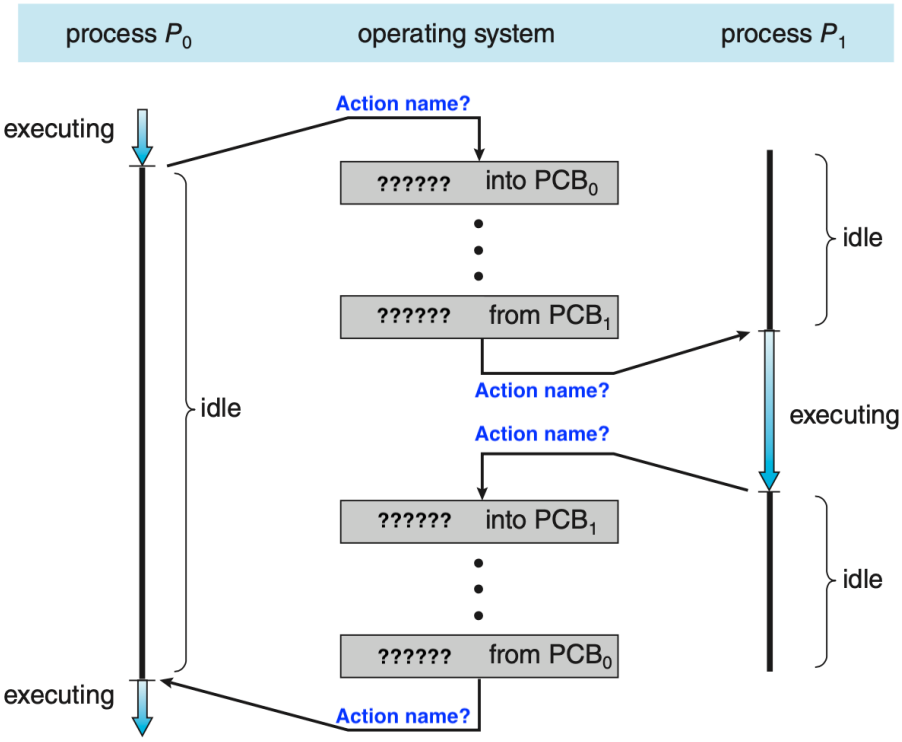
5. To manage running processes an operating system uses a Process Control Block (PCB). Describe the content in a PCB.



6. Processes pass a certain number of states during its life cycle. Describe these states, and the transitions between different states.



7. What operations are performed by an operating system in a context switch?



8. Discuss how the following pairs of scheduling criteria conflict in certain settings.

- CPU utilization **VS** response time
- Average turnaround time **VS** maximum waiting time

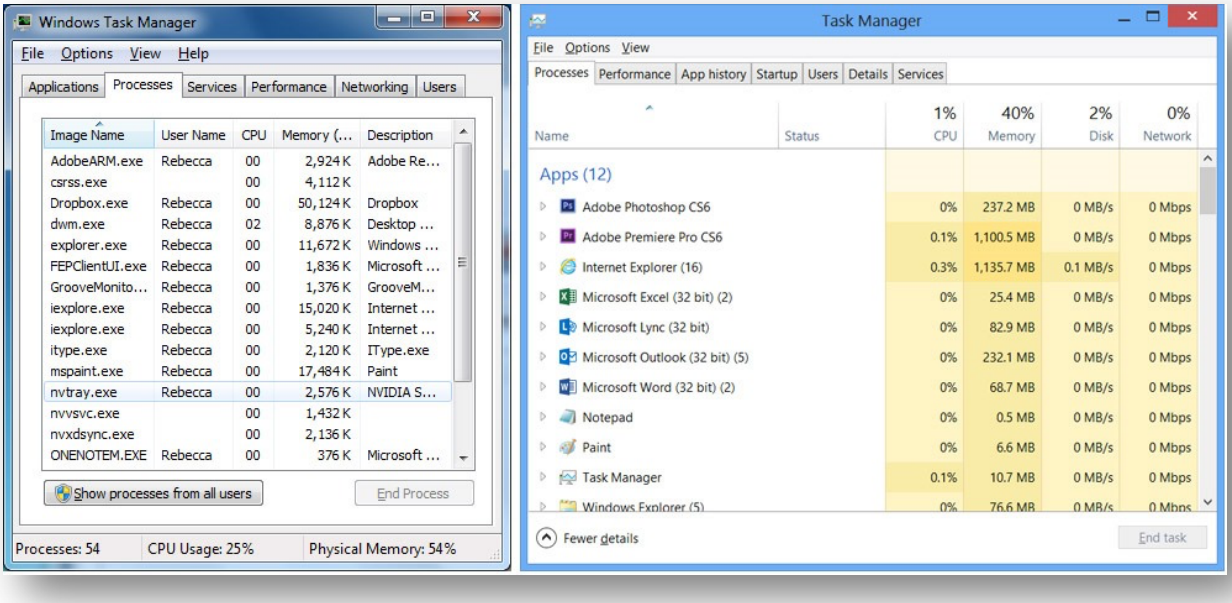
Kristianstad University
SE-291 88 Kristianstad
+46 44 20 30 00
www.hkr.se

9. Which of the following scheduling algorithms could result in starvation?

- First-come, first-served
- Shortest job first
- Round robin
- Priority

Part 2 – Task Manager

Option #0: Start the task manager in Windows. You can do this by pressing Ctrl+Alt+Delete and choose task manager in Windows, or by directly press Ctrl+Shift+Esc. Check the different alternatives in the menu to be comfortable with the different options. In specific, test the resource monitor. Then start a new application and see if you can find its process id in the list.



Option #1: Alternatively, if you are a Unix user (Linux/MacOS) you can open terminal and execute command: **ps aux** which will list all the processes currently running with all corresponding info about them.

```
USER               PID  %CPU %MEM      VSZ    RSS   TT  STAT STARTED      TIME COMMAND
smjeon           22206   8.5  3.3  5593024 550948   ??  S    11:22PM  14:53.06 /Applications/Google Chrome.app
smjeon           15516   4.4  0.2  6559572  26152   ??  S    금 10AM  43:27.17 com.docker.hyperkit -A -u -F vms
smjeon           21655   4.2  1.9  5606060 323036   ??  S    10:11PM   4:45.84 /Applications/Visual Studio Cod
smjeon             710   3.9  0.7  5760424 124572   ??  S    화 04PM  27:29.23 /Applications/Visual Studio Code
_windowserver      193   3.9  1.0  9243820 169264   ??  Ss   화 04PM 114:25.29 /System/Library/PrivateFramework
root                 1   2.3  0.1  4374344  15912   ??  Ss   화 04PM   7:20.33 /sbin/launchd
smjeon           21854   2.2  0.0  4334216   4924 s003  Ss   10:38PM   0:03.70 /bin/zsh -l
_coreaudiod        161   2.1  0.0  4359596   4100   ??  Ss   화 04PM   3:30.31 /usr/sbin/coreaudiod
smjeon             749   1.4  2.8  6008164 470604   ??  S    화 04PM  11:55.64 /Applications/Visual Studio Code
smjeon           10525   0.8  0.2  7310092  25364   ??  S    목 03PM   3:07.02 /Applications/MySQLWorkbench.app
smjeon            1489   0.5  0.0  4589448   8060   ??  S    화 09PM   1:11.53 /Applications/Google Chrome.app/
_hidd              104   0.5  0.0  4381920   5644   ??  Ss   화 04PM  40:58.07 /usr/libexec/hidd
smjeon           18914   0.4  1.3  5481316 223624   ??  S     1:38PM   2:02.92 /Applications/KakaoTalk.app/Con
smjeon           23297   0.4  0.7  4934884 112936   ??  S    12:29AM   0:32.37 /Applications/Visual Studio Cod
smjeon             797   0.3  0.0  4412816   7452   ??  S    화 04PM   1:19.74 /System/Library/CoreServices/Sir
_mysql              97   0.3  0.2  4938960  30116   ??  Ss   화 04PM  11:52.30 /usr/local/mysql/bin/mysqld --us
smjeon            1165   0.2  0.0  4913304   7860   ??  Ss   화 05PM   1:05.70 /System/Library/Input Methods/Ko
smjeon             742   0.2  0.2  4628740  39492   ??  S    화 04PM  10:44.82 /Applications/Google Chrome.app/
smjeon             366   0.1  0.0  4377656   4172   ??  S    화 04PM   0:39.04 /System/Library/PrivateFramework
root               103   0.1  0.0  4381016   7964   ??  Ss   화 04PM   2:33.40 /usr/sbin/bluetoothd
```

Part 3 – Processes and Threads (preparation for seminar 1)

1. Read about the class ProcessBuilder in the Java documentation.
https://docs.oracle.com/javase/8/docs/api/java/lang/ProcessBuilder.html

2. Create a new Java-application in your favorite IDE, e.g. IntelliJ. In the main method add the following code.

```
ProcessBuilder pb= new ProcessBuilder("calc");
try
 {
     pb.start ( );
 }
catch (IOException ex)
{
     Logger.getLogger
     (ProcessExercise.class.getName()).log(Level.SEVERE, null, ex);
 }
```

(The try and catch statement is included because the start method of ProcessBuilder will throw an exception if anything goes wrong.) Compile and run the code. What do you notice?

3. Change the command from “calc” to “calck”. What do you notice when you run the program?

4. Change the command to notepad and run the application. Change the command to “notepad”,”myfile.txt”. What observations do you note? Where is the file saved if you press save in notepad? (This is the reason for the filename “myfile.txt”. It is easier to find when we search if we choose a filename that we have not used on our computer.)

Kristianstad University
SE-291 88 Kristianstad
+46 44 20 30 00
www.hkr.se

5. Creating a new process is a good way when we would like an entire new application to start. However, in many cases we would like to run parts of an allocation concurrently. It can be done by creating new processes but it is often a very complex operation which uses a lot of system resources. A better approach is most often to create a new thread instead.

A good introduction to threads can be found in the tutorial at

http://docs.oracle.com/javase/tutorial/essential/concurrency/

You can follow the tutorial up to "The Simple Threads Example".

Part 4 - Scheduling Algorithms

When we consider different scheduling algorithms there are several different properties to study. Three of them are turnaround time, waiting time and response time. An explanation of each one of them can be given as followings:

Turnaround time: The time between the moment when a task arrives to the ready queue until it leaves the CPU and have finished its current CPU-burst.

Waiting time: The total time a task has spent in the ready queue. Note that for preemptive scheduling a task might be transitioned between the CPU and the ready queue until it finishes its CPU-burs. The waiting time can also be calculated as Turnaround time – CPU burst.

Response time: The time between arrival and the first time the task get access to the CPU.

In most situations we are most interested not in the performance for individual task, instead the average or in some cases the maximum time.

1. Consider the following set of processes, with of the length of the CPU burst given in milliseconds:

Process	Burst time	Priority
P1	2	2
P2	1	1
P3	8	4
P4	4	2
P5	5	3

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0.

a) Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF (nonpreemptive), non-preemptive priority, and RR (quantum=2).

Kristianstad University
SE-291 88 Kristianstad
+46 44 20 30 00
www.hkr.se

b) What is the turnaround time of each process for each of the scheduling algorithms in a?What is the waiting time of each process for each of the scheduling algorithms in a?

c) Which of the algorithms results in the minimum average waiting time (over all processes)?

2. Consider the following set of processes, with of the length of the CPU burst given in milliseconds:

Process	Burst time	Priority	Arrival time
P1	6	2	0
P2	12	1	1
P3	8	4	7
P4	4	2	11
P5	5	3	15

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, according to the above arrival times.

a) Draw six Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF (nonpreemptive), SJF (preemptive), non-preemptive priority, preemptive priority, and RR (quantum=2). (Assume a lower number of priority corresponds to higher priority.)

b) What is the turnaround time of each process for each of the scheduling algorithms in a?

c) What is the waiting time of each process for each of the scheduling algorithms in a?

d) Which of the algorithms results in the minimum average waiting time (over all processes)?

Fredrik Stridh
Deniel Alekseev