Exam Operating Systems
Tuesday April 11th 2017

- Please put your student identification card on your desk.
- Write your name, student number and the total number of sheets on the first sheet. Number the sheets!
- You may answer the problems in Dutch or in English.
- Please read each problem fully before solving it. Write neatly and carefully. If the handwriting is unreadable, or needs guessing to make something out of it, then the answer is rejected.
- The use of a (graphical) calculator is permitted.
- For answers without explanation (even if the answer is correct) no points are awarded.
- The exam consists of five problems. Each problem is worth the same amount of points.

Problem 1: General OS questions

- **(a)** What are the two main functions of an Operating System?
- **(b)** A UNIX process can be in one of the following three states: *running*, *ready*, or *blocked*. Describe two events that cause a process to transition from *running* to *ready*.
- **(c)** What are the four necessary and sufficient conditions for deadlock to occur?
- **(d)** Describe what a *context switch* is, and what the OS needs to do to implement it.
- **(e)** The *switch* routine is written in assembly. Describe another routine that is written in assembly.
- **(f)** CPU's have different privilege modes. Describe the two most used and how they are implemented in the operating system.

Problem 2: Unix system programming
Consider the following program:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

#define READ   0
#define WRITE  1
#define STDIN  0
#define STDOUT 1

int main() {
  int pid1, pid2, pfd[2];

  pipe(pfd);
  pid1 = fork();
  if (pid1 > 0) {
    pid2 = fork();
    if (pid2 > 0) {
      int status;
      close(pfd[READ]);
      close(pfd[WRITE]);
      waitpid(-1, &status, 0);
      waitpid(-1, &status, 0);
    } else {
      close(pfd[WRITE]);
      close(STDIN);
      dup(pfd[READ]);
      execl("/usr/bin/wc", "wc", NULL);
    }
  } else {
    close(pfd[READ]);
    close(STDOUT);
    dup(pfd[WRITE]);
    execl("/bin/ls", "ls", NULL);
  }
  return 0;
}
```

(a) Which UNIX shell command line does this program mimic?

(b) Explain how the program works, and explain the functionality of all the system calls that are used in the program.

(c) How many times will the following program print ”Hello world”? Explain your answer.

```
int main() {
  int i;
  for (i=0; i < 3; i++) {
    fork();
  }
  printf ("Hello world\n");
  return 0;
}
```

Problem 3: File Systems

(a) A filesystem of another computer is mounted under */mnt/remotefs*.
Consider the following command: *ln /mnt/remotefs/etc/passwd ~/passwd.*
What will happen and why? What would be a better way to do this?

(b) Where is disk-scheduling handled using modern rotational hard-disks? Which scheduling algorithm is most used today? Describe your answer using the following read request from *cilinder*: 16, 20, 80, 45, 50, 24, 36, 15, 70.

(c) We consider a (hypothethical) file system with only 16 blocks. A file system checker determines for each block how many times it is in use, and how many times it occurs in the free-list. The checker established the following counters:

Block Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
In use	1	1	0	1	0	1	2	1	1	0	0	1	1	1	0	0
Free	0	0	0	0	2	0	0	0	0	1	1	0	1	0	1	1

Are there file system errors in the file system? If yes, list the errors and explain how they should be repaired.

(d) A file has permissions *"0741"*. Explain what these permissions mean.

(e) Why is it that BIOS-based computers can only handle 4 Operating Systems on a hard-disk? What can you do to work around that limitation?

Problem 4: Virtual Memory

(a) Consider a system with only three physical memory frames that needs to handle the following page memory reference sequence:3, 2, 1, 0, 3, 2, 4, 3, 2, 1, 0, 4

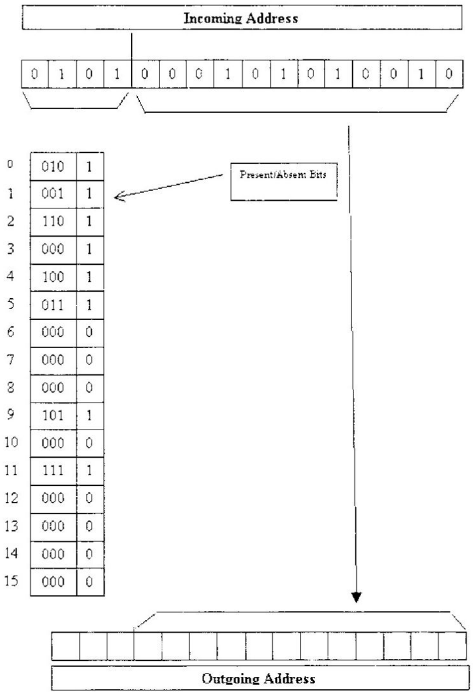
What is the number of page faults that would occur for each of the following page replacement strategies?

- Optimal page replacement algorithm
- LRU replacement algorithm
- FIFO replacement algorithm

(b) What will happen to the above if we add a fourth physical memory frame?

(c) In a *page table entry* there are more bits than *present/absent*. Describe 2 more bits and eleborate on their function.

(d) Consider a small virtual memory system with a single level page table (see figure). The system uses 16-bit addresses and 4KB (i.e. 4096 bytes) pages. Note that a page is loaded in physical memory if its *Present/Absent* bit is set to 1.



0	010	1
1	001	1
2	110	1
3	000	1
4	100	1
5	011	1
6	000	0
7	000	0
8	000	0
9	101	1
10	000	0
11	111	1
12	000	0
13	000	0
14	000	0
15	000	0

- A process wants to read (virtual) address 20818 (binary 0101000101010010). Explain in detail what will happen.
- A process wants to read (virtual) address 28680 (binary 0111000000001000). Explain in detail what will happen.

(e) Describe the function of the *Translation Lookaside Buffer*. What is the typical location of this device?

Problem 5: Scheduling

(a) Five *batch jobs* A, B, C, D, E are simultaneously offered to an operator of some batch processing facility of a computer center. The expected execution times of the jobs are respectively 60, 40, 15, 20, and 30 hours. In which order should the operator start the jobs in order to *minimize* the *average turnaround time*, and what is the value of this minimum?

(b) Given are four processes that 'arrive' (are submitted) at different times. All processes are completely CPU bound. Times are denoted in seconds. Of each process, the time of arrival and its execution time are given in the following table. The operating system keeps track of a set of processes that are in the *Ready*-state. If at a certain time processes arrive and also a context switch takes place (in case of pre-emptive scheduling), then first the newly arrived processes are added to the Ready-set before the scheduler adds the active process to it and decides which process to run next.

process	arrival time	execution time
A	0	4
B	2	9
C	3	3
D	5	5

Determine the order of execution of the processes and the average *waiting time* for each of the following systems.

- System with non-preemptive *First Come First Served (FCFS)* scheduling
- System with non-preemptive *Shortest Job First* (SJF) scheduling
- System with pre-emptive *Shortest Remaining Time Next* (SRTN) scheduling with a time quantum of 2 seconds
- System with preemptive *Round-Robin Scheduling* with a time quantum of 2 seconds

(c) We now consider priority scheduling with three priority levels. The above processes have the following priorities:

- highest priority: processes B and D
- middle priority: process C
- lowest priority: process A

Determine the order of execution of the processes and the average *turnaround time* for a system with pre-emptive priority scheduling with a time quantum of 1 second. Assume that each priority class is scheduled in round robin fashion.

(d) In an interactive system the round-robin scheduler is configured with a quantum of x. What would be the best size of x? Explain your decision.