| **Department of Computer Science** | **Operating Systems** |
|---|---|
| **VU University Amsterdam** | **09.02.2017** |

***This exam consists of two pages. No calculator, pencil, or open books allowed. Concise answers!***

*1* Which operating system (OS) architecture would you opt for in each of the following scenarios?

(a) Limit the damage a software bug in a single module can do to the rest of the OS.

(b) Maximize OS performance and compatibility with commodity applications.

(c) Enforce the principle of least authority to improve security in the OS.

(d) Optimize OS behavior for each individual application.

*10pt*

*2* Which of the following instructions should be allowed only in kernel mode? Mark all the correct answers.

(a) Disable all interrupts.

(b) Flush TLB.

(c) Trap instruction.

(d) Read the time-of-day clock.

(e) Set the time-of-day clock.

(f) Change the memory map.

*10pt*

*3* How does the OS ensure that different threads in the same process share the address space and that different processes do not share it? Are there exceptions to these abstractions? Say a region of the address space that can be shared across processes and a region of the address space that is private to a thread? *10pt*

*4* Consider a real-time system with four voice calls of periodicity 5 msec each with CPU time per call of 1 msec, and one video stream of periodicity 33 ms with CPU time per call of 11 msec. Is this system schedulable? What is the maximum number of voice calls for this system to be schedulable with a single video stream? *10pt*

*5* What is the difference between a mutex and a spin lock? When would you use one or the other? Would it make sense to introduce a synchronization primitive that behaves like a spinlock at first and resorts to mutex-like behavior if unsuccessful for an extended period of time? *10pt*

*6* You run the program below on a system with 4 MB of RAM and a virtual memory subsystem using a LRU page replacement algorithm. As you run the program, you observe “thrashing” (i.e., the program constantly page faulting) and thus very poor performance. What happened? How would you change the virtual memory subsystem to fix the problem? *10pt*

```
1  extern void init_array(char* array);
2  extern void print_char(char c);
3
4  char array[4*1024*1024];
5
6  int main(int argc, char **argv)
7  {
8    int i, j;
9    init_array(array);
10
11   for (i=0; i<1000000; i++) {
12     for (j=0; j<1024; j++) {
13       print_char(array[4*1024*j]);
14     }
15   }
16
17   return 0;
18 }
```

*7* What is copy-on-write (COW) and how does it work? How is COW implemented in a paging system? Also, give one use case for COW. *10pt*

*8* You are given a program reading a single byte from a file in a UNIX-like file system (with inodes using up to triple indirect blocks). You profile the I/O disk operations performed during two consecutive runs of the program and observe the following results on 2 different systems (1, 2):

1. First run: 5 reads and 1 write, second run: 0 reads and 0 writes.
2. First run: 5 reads and 1 write, second run: 0 reads and 1 write.

Explain the nature of such disk operations and that of the differences observed on the 2 systems. *10pt*

*9* Disk requests come in to the driver for cylinders 10, 22, 20, 2, 40, 6, and 38, in that order. A seek takes 10 msec per cylinder moved. How much seek time is needed for:

(a) First-come, first served.

(b) Closest cylinder next (Shortest Seek Time First).

(c) Elevator algorithm (initially moving upward).

(d) Modified elevator algorithm (moving upward, the lowest cylinder processed right after the highest one).

(e) Random.

In all cases, the arm is initially at cylinder 20. *10pt*

*10* Give an example of a deadlock taken from politics. Mention two mechanisms to handle the deadlock. *10pt*