| **Department of Computer Science** | **Operating Systems** |
| **VU Amsterdam** | **04.02.2019** |

This exam consists of one page. No calculator, pencil, or open books allowed. Concise answers!

1 By exploiting a bug in the code, an attacker has taken over the execution of a printer driver instance and seeks to corrupt as much operating system state (e.g., process table, i-node table, etc.) as possible. Explain what state the attacker is able to corrupt, and why, in the following operating system organizations: monolithic kernels, microkernel-based client-server systems, exokernels. *10pt*

2 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? What about a primitive that behaves like a mutex at first and resorts to spinlock-like behavior if unsuccessful for an extended period of time? *15pt*

3 Consider the C code below. On a typical OS implementation, executing the `NULL` pointer dereference at line 7 would trigger an exception (segmentation fault). How can an OS implementation based on virtual memory enforce this behavior? In absence of virtual memory, can an implementation based on a memory abstraction using only one `base` and one `limit` register per running program enforce this behavior? And, if so, how? *Hint*: Think about what memory location is accessed on every `NULL` pointer dereference. *15pt*

```
1  #define NULL 0
2  #include <stdio.h>
3
4  int main()
5  {
6    int *number = NULL;
7    printf("Hello, world number %d!", *number);
8    return 0;
9  }
```

4 Suppose that a machine has 38-bit virtual addresses and 32-bit physical addresses.

   (a) What is the main advantage of a multilevel page table over a single-level one?

   (b) And what are the advantages and disadvantages of a multilevel page table organization compared to inverted page tables?

   (c) With a two-level page table organization, 16-KB pages, 4-byte entries, and each page table stored in a single page, how many bits should be used to index the top-level page table and how many bits to index the second-level page table at each page table walk? Explain. *15pt*

5 It has been suggested that the first part of each UNIX file be kept in the same disk block as its i-node. What good would this do? And, if this is beneficial, why not simply store the entire file next to its i-node? *15pt*

6 Describe the steps a device driver needs to perform to read a block of data from a particular storage device. Briefly describe all the alternatives to complete the request (polling, interrupts, interrupts + DMA) and mention one compelling use case for each alternative. *15pt*

7 Why is the Banker's algorithm rarely used in practice? Are there more practical alternatives? Support your answer with an example. *15pt*