

# Operating systems 2018

Operating systems (1DT044)  
Operating systems and process oriented programming (1DT096)  
Uppsala university

## Seminar questions

### Module 2

The process concept and inter process communication  
File descriptors, standard streams, and I/O redirection

Document last updated  
2018-01-25

#### The process concept

1. What is meant by a process?
2. A process needs at least two critical resources, name these resources.
3. Name and describe the various memory segments used by a process.
4. A process can be in different states. Name and explain the purpose of each such state.
5. Draw a diagram showing how the process states are related using directed arrows showing possible state transitions.

#### Process control block (PCB)

6. What is the purpose of the PCB?
7. Give examples of data stored in the PCB.
8. Explain how PCBs can be used to construct various process queues.

#### fork

9. What is the purpose of the fork system call?
10. What do we mean with parent and child?
11. How many times does fork return?
12. What are the possible return values of fork?
13. After calling fork, how can the program know if it is executing in the parent or in the child?

#### exit

14. What is the purpose of the exit system call?

## **exec**

15. What is the purpose of the exec family of system calls?
16. When calling a function or invoking a system call, normally execution will return back to the caller, possibly with a return value. Is this true for the exec family of system calls? Justify your answer.

## **wait**

17. What is the purpose of the wait system call?

## **Zombie processes**

18. What is the purpose of the zombie process state? When does a process become a zombie?

## **Signals**

19. What is the purpose of signals?
20. What are the limitations of signals?
21. What happens when a process receives a signal?

## **File descriptors**

22. Explain the file descriptor concept.

## **Pipes**

23. What is a pipe?
24. How are file descriptors used together with pipes?
25. How do we create a pipe? What is the result of creating a pipe?
26. How can we make two processes share a pipe in a producer-consumer manner?
27. What happens if we read from an empty pipe and there are a) open write descriptors attached to the pipe, or b) no open write descriptors attached to the pipe?
28. What happens if we write to a) a full pipe if there are open read descriptors attached to the pipe, or b) a pipe with no open read descriptors attached to the pipe?

## **dup2**

29. What is dup2 doing to file descriptors?
30. How can this be useful?

## **Random mystery**

31. In C, the `rand` library function can be used to generate pseudorandom numbers. How is it possible for `rand` to return different values on consecutive calls?
32. A parent process calls `srand` to seed the pseudo random generator (PRNG) and then uses `fork` to create a number of child processes. Each child generates a sequence of random numbers by calling `rand`. Can you make any predictions about the sequences? Justify your answer.