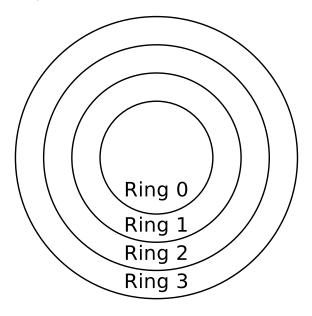
Exercise Sheet 4

Exercise 1 (System Calls)

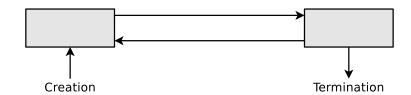
1. x86-CPUs contain 4 privilege levels ("rings") for processes. Mark in the diagram (*clearly visible!*) the kernel mode and the user mode.



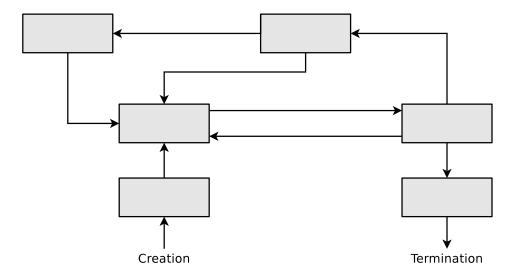
- 2. Which ring contains the kernel of the operating system?
- 3. Which ring contains the applications of the users?
- 4. Processes of which ring have full access to the hardware?
- 5. Name a reason for the differentiation between user mode and kernel mode.

6. W	That is a system call?
7. W	That is a context switch?
	Tame two reasons why user mode processes should not call system calls irectly.
	That alternatives exist, if user mode processes should not call system calls irectly?
Exer	cise 2 (Processes)
	Which three sorts of process context information stores the operating system?
	Which process context information are not stored in the process control lock?
3. W	Why does the process control block not store all process context information?
	ist all information stored in the process control block of a RIOT process hread)? (Check at https://doc.riot-os.org))

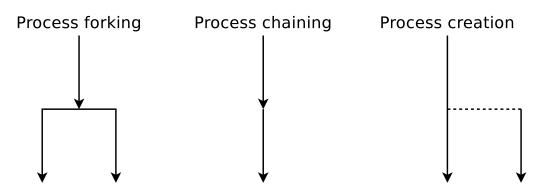
- 5. What is the task of the dispatcher?
- 6. What is the task of the scheduler?
- 7. The process state model with 2 states is the smallest possible process model. Enter the names of the states in the diagram of the process state model with 2 states.



- 8. Does the process state model with 2 states make sense? Explain your answer shortly.
- 9. Enter the names of the states in the diagram of the process state model with 6 states.



- 10. What is the task of the process table?
- 11. How many status lists for processes in "blocked" state manages the operating system?
- 12. Describe the effect of calling the system call fork().
- 13. Describe the effect of calling the system call exec().
- 14. The three diagrams below show all existing ways of creating a new process. Specify for each diagram, which system call(s) are required to implement the illustrated way of process creation.



15. A parent process (PID = 75) with the characteristics, described in the table below, creates a child process (PID = 198) by using the system call fork(). Enter the four missing values into the table.

Parent Process	Child Process
72	
7.5	100
75	198
18	
	72 75

16. Describe what init is and what its task is.

17. Name the differences of a child process from the parent process shortly after its creation.

18. Describe the effect, when a parent process is terminated before the child process.

check should be printed out.

19. Describe what data the Text Segment contains. 20. Describe what data the Heap contains. 21. Describe what data the Stack contains. Exercise 3 (Time-based Command Execution, Control Structures, Archiving) 1. Program a shell script, which reads two numbers as command line arguments. The script should check whether the numbers are identical, and print out the result of the check. 2. Extend the shell script in a way that if the numbers are not identical, it is checked, which one of the two numbers is the larger one. The result of the 3. Program a shell script, which creates a backup of a directory of your choice. The script should create an archive file with the file extension .tar.bz2 from the directory. The archive file should be stored in the directory /tmp. The name of the archive file should correspond to the following naming scheme:

Backup_<USERNAME>_<YEAR>_<MONTH>_<DAY>.tar.bz2

The fields <USERNAME>, <YEAR>, <MONTH> and <DAY> should be replaced by the current values.

4. Program a shell script, which checks if already today an archive file was created according to the naming scheme of subtask 3. The result of the check should be printed out in the shell.

5. Write two cron jobs. She first cron job should execute at 6:15 am on every day (except on weekends) the shell script from subtask 3, which creates the archive file with the backup. She second cron job should execute at 11:45 am on every day (except on weekends) the shell script from subtask 4, which checks, whether already today an archive file was created. The output from the shell scripts should be appended to a file /tmp/backup.log. If the archive file Backup...tar.bz2 has been created successfully, this should be noted in the log file /tmp/backup.log. Before each new entry in the file, lines according to the following pattern (with current values) should be inserted into the log file /tmp/backup.log.

20.11.2013 --- Time: 21:39:51

Content: Topics of slide set 5 Page 8 of 12

Exercise 4 (Shell Scripts)

- 1. Program a shell script, which checks for a file, which is specified as an argument, whether it exists and if it is a file, a directory, a symbolic link, a socket or a named pipe.
 - The script should print out the result of the check.

Content: Topics of slide set 5 Page 9 of 12

2. Extend the shell script from subtask 1 in a way that if the file, which is specified as an argument, exists, it is checked, if the file could be executed and if write access would be possible.

- 3. Program a shell script, which reads so long text on the command line, until it is terminated by typing END.
 - The script should convert the text, which is read in from the command line, to uppercase.

4. Program a shell script, which prints out the number of running processes for all logged in users.

- 5. Extend the shell script from subtask 4 in a way that that the output is sorted.
 - The user with most processes should stand at the beginning.

- 6. Program a shell script, which checks after start every 10 seconds, if a file /tmp/lock.txt exists.
 - Each time after the script has checked the existence of the file, it should output an appropriate message on the shell.
 - Once the file /tmp/lock.txt exists, the script should terminate itself.

Content: Topics of slide set 5 Page 12 of 12