A+ will be down for a version upgrade on Tuesday 03.01.2023 at 9-12.

**This course has already ended.**

« 7.1 Synchronization (/os/2022/materials_... 8. Linux shell » (/os/2022/materials_m08/)

CS-C3140 (/os/2022/) / 7. Synchronization and deadlocks (/os/2022/materials_m07/) / 7.2 Deadlocks

# Deadlocks

# Exercise 1

⚠ The deadline for the assignment has passed (Sunday, 27 November 2022, 23:59).

## Deadlocks

**1. Review questions. Please consult Chapters 5 & 6 from "Operating Systems - Internals and Design Principles" by William Stallings**

**Question 1** 3 / 3

**What are the conditions(some/all) that must be present for deadlock to occur?**

- [x] **mutual exclusion**
- [x] **no pre-emption**
- [x] **circular wait**
- [x] **hold-and-wait**

✔ Correct!

**Question 2** 2 / 2

**How can the hold-and-wait condition be prevented?**

- [ ] by requiring that a process request all of its required resources at one time, and blocking all the other processes until all requests for the first process can be granted
- [ ] by requiring that a process request at least half of its required resources at the start, and blocking the process until all requests can be granted simultaneously

- [x] **by requiring that a process requests all of its required resources at one time, and blocking the process until all requests can be granted simultaneously**

✔ Correct!

**Question 3** 3 / 3

**Why you cannot disallow mutual exclusion in order to prevent deadlocks?**

- [x] **there are some resources (like printers) that are inherently non-sharable, and it is impossible to disallow mutual exclusion**
- [ ] some resources, such as files, should have exclusive access for writes even if reads are not exclusive
- [x] **if mutual exclusion is disallowed, then all non-sharable resources become sharable**

✔ Correct!

**Question 4** 2 / 2

**How can the circular wait condition be prevented?**

- [ ] by having one process use the resources it needs before allocating others
- [x] **by defining a linear ordering of resource types**
- [ ] by grouping processes into different categories

✔ Correct!

**Question 5** 2 / 2

**Which of the methods may be adopted to recover from deadlocks.**

- [x] **Abort all deadlocked processes**
- [x] **Successively abort deadlocked processes until deadlock no longer exists**
- [x] **Successively preempt resources until deadlock no longer exists**
- [x] **Back up each deadlocked process to some previously defined checkpoint, and restart all processes.**

✔ Correct!

Submit

« 7.1 Synchronization (/os/2022/materials_...

8. Linux shell » (/os/2022/materials_m08/)