NTNU | Norwegian University of Science and Technology

Operating Systems

Q&A Session – 18.02.2021

Michael Engel

Corona situation and consultation

- The Gløshaugen campus is open again…
    - but only for small groups as before (adhere to the known Corona precautions as usual when you're on campus!)
    - No physical lectures for the foreseeable future

- Piazza license payment finally confirmed…
    - …but the nag requester on piazza hasn't disappeared yet :(

- Request for TA consultation hours
    - We are organizing this and will offer at least two hours per week (if all goes well, four hours)
    - A TA will be available in a zoom session
    - You can join from home or from campus if you are doing group work there
    - Details (times etc.) early next week

Practical exercise organization

- Submission deadline for PE2 ***extended***
  - …due to popular demand! 😉
    - There was some confusion since we postponed the submission of PE1
  - New deadline:
    tomorrow, Friday, February 19th 14:00

- **The following PEs** will (most probably) not be postponed
  - So for TE3, PE3 and all following ones the deadline is:
    Thursday at 12:00

- Some problems with getting correct points in PE1
  - Two groups accidentally received 0 points, looking into it

PE2

Confusion on some details:

- Error checking – where, how, to which extent?
  - **scanf** error checking is important
    - scanf tells you if e.g. the input was a character string instead of a number
  - **printf** error checking is less important
    - printf returns the number of successfully printed characters
    - this can be < the number of characters to print
      - …but rarely on a terminal
      - rather when writing to a file and the disk is full
  - **exit** error checking? `void exit(int status);`

```
"RETURN VALUES
     The exit() and _Exit() functions never return."
```

PE2

Confusion on some details:

- **wait and zombies**
  - Most of you have figured out that wait is problematic:
    "The wait(2) system call suspends execution of the calling thread until one of its children terminates."
    - This would keep the program from reading an additional alarm time and starting a new child proc.
  - Read the manpage 😎
    - waitpid(2) might also help:
      "If pid is -1, the call waits for any child process"
    - Tip: Look at the "WNOHANG" option…
- **"Why don’t we use signal?"**
  - Because we didn’t discuss Unix IPC and signals in the lecture so far (but great that you have found it!)