Department of Computer Science and Engineering
Chalmers

Exam in Unix Internals EDA203/DIT681

DAY: 2012-08-31 TIME: 8.30-12.30 PLACE: V building
==========

Examiner: Arne Dahlberg

Questions during exam: Arne Dahlberg, 160649, 772 1705

Solutions: No solutions will be posted

Grading Policy: EDA203 3: 30-38, 4: 39-47, 5: 48-60
DIT681 G: 30-47, VG: 48-60

Aids during the exam:

- McKusick, Neville-Neil: The design and implementation of the FreeBSD operating system.
- An English language dictionary.

Instructions:

- Start answering each assignment on a new page; number the pages and use only one side of each sheet of paper.
- Justify all answers. Lack of justification can lead to loss of credit even if the answer might be correct.
- No references to the text book is allowed and copying of text from the book is not allowed.
- If you make any assumptions in answering any item, do not forget to clearly state what you assume.
- Write clearly! If I cannot read your solution, I will assume that it is wrong.

Problem 1 (10p)

a. Describe how the old historic thread scheduler in FreeBSD works. (6p)

b. Which mechanism in the scheduler prevents a CPU-intense process from disturbing interactive processes too much? (2p)

c. Why is the priority for sleeping processes not periodically updated? (2p)

Problem 2 (10p)

a. Describe how a process group works. What is the most important reason why process groups were added to the system? Why is process groups collected into a session? Explain how a a process group can become an "orphaned process group" ? What will happen to the processes that are members in such an "orphaned process group" ? (7p)

b. A login-shell starts a process in the background and then exits (logging out). Explain the mechanism that prevents the remaining background process to read from the terminal. (3p)

Problem 3 (10p)

a. When a new process is created a copy of the old process should be created by the *fork* system call. To copy a big process will however take a long time. In the FreeBSD implementation of *fork* this copying is avoided. Describe how copying is avoided in FreeBSD *fork* (not vfork!). (5p)

b. When a process is terminated (by exit) all its pages are not always returned to the free list. Give two reasons why pages may not be returned to the free list. (5p)

Problem 4 (10p)

a. When a file is opened by the open system call, a path name have to be converted to a vnode pointer. Describe how this translation is done. (5p)

b. One use of vnodes is to direct a system call to the correct file system code. Explain in detail how this works for a *read* system call. (5p)

Problem 5 (10p)

a. In FreeBSD there is a so called SYN-cache. What is the reason that this cache exists? (2p)

b. Describe how a new TCP connection is set up using the SYN-cache. (5p)

c. The TCP protocol use several timers to protect against lost packages. One such timer is the *persist timer*. When is the *persist timer* used and for what reason? (3p)

Problem 6 (10p)

Assume that a one kilobyte UDP datagram has been sent by some process to a remote location at the Internet. Describe the handling of this packet from the call to the *sosend()* routine till it has been sent by the local Ethernet interface. Assume that the Ethernet destination address can be found in the ARP cache. The description shall include where different headers are added and the calls between the different protocol layers. (Description of handling at transport, ne twork and link layer is needed)