Create a Custom Syscall

Create a custom syscall to the x86 64 bit architecture that tells you the current state of the system.

What to Submit

- Implementation of the new syscall as a series of patches
- Source code to a userspace utility that tests the functionality of the new created syscall
- Time spent on assignment

Procedure

- Clone Linuses git tree.
- Checkout release version 6.16
- Checkout out own branch
- Implement a new syscall for x86 64 bit architecture
- Write a user space utility that will prove the functionality of the new syscall
- Subject of the mail should have the form
    - KDLP25 E02 VX
    - Where X is the number of the revision of the email
    - Revision number starts at “1”

Requirements

- Number of the new syscall is 600 (600 decimal)
- The syscall will take one numerical parameter as an input
- The syscall will return one numerical parameter as an output
- If the input of the syscall is 1, the syscall will return count of currently running processes
- If the input of the syscall is 2, the syscall will return count of currently loaded modules
- If the input of the syscall is 3, the syscall will return count of filesystems, that require a block device
- If the input of the syscall is any other number a constant -1 will be returned
- The patches for the syscall need to pass the checkpatch.pl script
- Patches will be generated using git format-patch from your patches, with each patch describing your changes and a cover letter explaining the whole project

Hint: All of system information can be checked in the /proc directory from the user space

Points

- Maximum points for this assignment are 20.