



HW4 – System call

תאריך הגשה: 21.06.2023 בשעה 23:59

נהלי פנייה: בכל שאלה הנוגעת לתרגיל, אנא כתבו בפורום הייעודי במודל. פניות הנוגעות להארכה בהגשת תרגיל הבית – אנא כתבו מייל מפורט למתרגל האחראי של הקורס בצירוף אישורים מתאימים. מייל: yonatanko@campus.technion.ac.il

Your task.....	1
How we test.....	2
Submission.....	2
Where to start	3
System calls in xv6	3
Preparing for XV6 system calls assignment.....	3
analyzing source code for homework	3

In this assignment you will add a new system call to the XV6 operating system.

This exercise will show you how function calls are passed between user space and kernel space.

The next assignment will use this one, so make sure you complete it and understand how the internal mechanism of the OS is working.

Before you start: you need to have the XV6 system installed and make sure it runs.

This is covered in the document "Installing XV6".



Your task

Add a command called `lsproc` that will print the pagetable address for all running processes.

- The output must be sorted by **ascending process id**
- For each found process, print these columns:
name, pid, state, pagetable address

The steps are:

1. Edit some files (your goal is to understand which files).
2. Activate the XV6 terminal with the following command: `make qemu-nox`.
3. If everything went well, you would see `xv6$` prompt in a new line

```
Booting from Hard Disk..xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
xv6$ █
```

4. Type the `lsproc` command
5. The following output is expected:

name	pid	state	address
init	1	SLEEPING	8dfbb000
sh	2	SLEEPING	8df73000
lsproc	3	RUNNING	8df23000



How we test

Your code will be fed into automatic code tester. The output will be compared to a reference output.

- Hint: In xv6, commands (in user space) are run by the shell.
You need to write a **user space program** that will call the new system function and print the results.
- The new system call you will create bridges the worlds of user and kernel spaces, and therefore involves some 'magic' and affects several files. The magic is done by manipulating the stack, triggering an interrupt, and changing internal processor state. Luckily, we don't need to understand exactly how it works – just look for existing commands.

Submission

1. Run make clean to delete the compiled files.
 - a. Create **patch file** containing all your changes and only them.
name it HW_04_ID1_ID2.patch
2. Check yourself here:
<http://homework-checker.eastus.cloudapp.azure.com/94210/submit/hw/4>
3. Upload the patch to Moodle
4. How to create a patch? Follow the instructions given in the following video:
<https://panoptotech.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=82d8dea8-7cea-4c0c-9732-abc800f453bb>



Where to start

Watch the following videos.

The videos were recorded last years, and there are minor differences between the video and the text. It is still helpful!

Watch the videos in the order they are listed here.

System calls in xv6

<https://panoptotech.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=1dca50f0-c769-4696-919d-aba3009313d4>

Preparing for XV6 system calls assignment

<https://panoptotech.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=a7d95740-b090-4ee7-9d31-abbf00e47f52>

The relevant part for this exercise starts at minute 47:30.

Up to this point, the focus is on how to work with the debugger (optional).

Please note that the task in the year the video was taken is different than this year's task, but the essence of the exercise is the same - there is no need to reinvent the wheel. You need to search in the web for the relevant material, understand which files need to be edited, and write some code lines (there aren't many of them).

analyzing source code for homework

<https://panoptotech.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=756e4296-1ad6-4e79-804a-abc100f606f7>