4segmentation-lab

CST 334 (Operating Systems)
Dr. Glenn Bruns

Lab: Segmentation

1. Copy these files to a directory of your own on mlc104:

    ○ /home/CLASSES/brunsglenn/OSTEP/HW-segmentation/segmentation.py

    ○ /home/CLASSES/brunsglenn/OSTEP/HW-segmentation/README-segmentation

or download it from the textbook site:

pages.cs.wisc.edu/~remzi/OSTEP/Homework/homework.html

2. Run the program without any options:

```
$ ./segmentation.py
```

Follow the directions that are output. For each virtual address, compute the physical address, or state that there is a segmentation violation. There are only two segments.

For example, the second address you see in the virtual address trace is 0x0000019e. The size of the virtual address space is 1k, which means a 10 bit address (since 2^10 = 1024). The left-most of the 10 bits is the segment bit. Look at the rightmost 3 hexadecimal digits of the virtual address:

19e

which in binary is

00**01 1001 1110**

The segment bit (in red) shows this is segment 0, not segment 1. To translate to a physical address

- check that the offset (in blue, decimal 414) is less than the bounds value for segment 0 (which is 472 decimal). 414 < 472, so no segmentation violation.

- if no segmentation violation, add the offset to the base value for segment 0 (which is 6890 decimal). This gives physical address 414 + 6890 = 7304, which is the answer.

Do this for the other addresses yourself. Remember, addresses in segment 1 grow backwards.

When you're done, check your answer by running the program again, but with the -c option.

3. Do question 1 at the end of OSTEP chapter 16. It is further work with segmentation.py

4. Do question 2 at the end of chapter 16.

5. If you still have time, do question 3 at the end of chapter 16.