CSE 4/521 - Introduction to Operating Systems
Summer 2018
Homework 3

Objective:

1. This homework is designed to analyze the reading comprehension ability of a student in browsing through and understanding the PintOS source code – more specifically, in terms of Project 2.
2. Team submission is advocated in-order to promote better communication of technical concepts among team members, develop an intuition of proper division-of-tasks, and be responsible for each other in a group setting.

Total Points: 60
Submission deadline: Monday- Jul 30, 2018 (Inclass)
Submission guidelines: One submission per team
Grading criteria: All members of the team receive the same score
You can (add/negate) 5 homework buffer days of this semester towards the submission of this homework.
Please note the academic integrity policy at: http://academicintegrity.buffalo.edu/policies

To answer a part of these questions, please refer the extensive documentation at: https://web.stanford.edu/class/cs140/projects/pintos/pintos.html (more specifically, project 2 – “userprograms”). Reading through the source code and discussing these questions with your team members would be also be necessary to answer the questions. The ‘reading comprehension ability’ of the userprogram source code is essential before beginning Project/Checkpoint 2.

1. In PintOS, how many tests pass/fail under “userprog” directory? **(1)**
2. What is the kernel memory region range in PintOS? **(1)**
3. What is “functionptr” in some of the test cases for Project 2? **(1)**
4. Mention 3 ways how to break a system call in PintOS? Give examples. **(3x2 = 6)**
5. How is this statement used in PintOS? **(2)**
   ```
   static char dst[8192] __attribute__ ((section (“.testEndmem,\”aw\”,@nobits# ”)));
   ```
6. List at least 3 “bad pointers” you encountered in the source code of tests of userprograms. Mention which file is used and for what purpose. **(3x2 = 6)**
7. What are `.dat` files in PintOS test cases? **(1)**
8. Name all the syscall implementation that are checked in Project 2. **(5)**
9. Mention all 7 tests that check for Robustness of system call. In 1 sentence summarize the unique characteristics of each one of them. **(7x2 = 14)**
10. What is ELF? What is ELF program header and ELF executable header? What’s the difference between them? **(1+1+1 = 3)**
11. What’s the difference between the pointers `*EIP` and `*ESP` in PintOS? What values are they usually initialized to? Which regions of the memory do they point? **(1+1+1=3)**
12. What are the different functions performed by `load()` in `process.c`? What is the purpose of each one of them? For all the functions you stated, explain what would happen if that function is removed. (Commenting out and running through GDB might help). **(5)**
13. What is `PANIC`ing the kernel? **(1)**
14. Why is `page0` not allowed to be mappable? **(2)**
15. Why is 80x86 architecture in PintOS considered segmented? What is TSS? Why should you not change anything in TSS? **(1+1+1 = 3)**
16. What is the purpose of ‘examples’ directory? **(2)**
17. Which system utility would you use to check correct argument passing in userprograms. **(1)**
18. How does PintOS (not C in general) handle argument passing in userprograms? **(3)**