Department of Computing

CS 330: Operating Systems

BSCS: 6AB

Lab 2: **Introduction to Pintos Programming Project 1**

CLO3(Design & implement various pieces of OS software)

Date: 26-9-2018

Time: 9am-12pm and 2pm-5 pm

Instructor: Dr. Fahad Javed

Lab 3: Introduction to Pintos Programming Project 1

Introduction

This lab is designed to be a review and help session to get you started on the first Pint OS programming project.

Objectives

Objective of this lab is to entertain your quires related to the first Project of Pint OS and check your understanding about the starter code.

Tools/Software Requirement

Pint OS

Description

In this lab we will cover the following three issues:

i) Understanding the requirements for the first Pintos programming project.

ii) Understanding the starter codebase that you have to modify to implement the first Pintos programming project.

iii) Help with any issues installing Pintos. If you have any questions or problems, please talk to the course staff and we will help you fix them.

Installation:

You can download Pintos from here:-

http://web.stanford.edu/class/cs140/projects/pintos/pintos.tar.gz

Additional Instructions to run Pintos in your own Ubuntu VM:

i) Install qemu. To do this, simply run the following command in Ubuntu Linux:

sudo apt-get install qemu

ii) Create a symbolic link for the qemu binary /usr/bin/qemu-system-i386, so that Pintos can actually find and use it:

sudo ln -s /usr/bin/qemu-system-i386 /usr/bin/qemu

If your qemu installation includes the binary /usr/bin/qemu-system-x86_64, then use that instead for better results:

sudo ln -s /usr/bin/qemu-system-x86_64 /usr/bin/qemu

iii) Assuming you installed Pintos in $PINTOS_HOME, in $PINTOS_HOME/src/Make.config, add the following near the end of the file, right after the line "DEPS = -MMD -MF $(@:.o=.d)":

Turn off default optimisation in GCC 4.6

CFLAGS += -fno-omit-frame-pointer
iv) In $PINTOS_HOME/utils/pintos, change line number 103 that says:
$sim = "bochs" if !defined $sim;
to:
$sim = "qemu" if !defined $sim;
v) Change the last line of $PINTOS_HOME/src/threads/Make.vars to:
SIMULATOR = --qemu
vi) Add $PINTOS_HOME/src/utils to your PATH environment variable. For example, if
you installed Pintos in /home/student/pintos, you should add
/home/student/pintos/src/utils/ to your PATH environment variable. You can
do this as follows:
--
export PATH=$PATH:/home/student/pintos/src/utils/

Problem with Newer Versions of Qemu:

The newer version of qemu has a different shutdown sequence than the older version. As a result, while
your code may work correctly, qemu may not exit properly, cause 'make check' to fail the test.
To solve this problem, you need to add a single line to devices/shutdown.c. After line number 104 in the
function shutdown_power_off( ), add the following line:
outw (0xB004, 0x2000);
So your new code at that segment will look like:
for (p = s; *p != '\0'; p++) {
outw (0xB004, 0x2000);
outb (0x8900, *p);
}

Pintos Quick Start

While there are detailed instructions to use Pintos on its webpage
(http://web.stanford.edu/class/cs140/projects/pintos/pintos_1.html#SEC1), you might find them
hard to follow at first. So here is a series of steps that will allow you to quickly start on Pintos

project # 1. Note that you first have to download and extract Pintos, run the additional instructions above, and then do the following:

i) cd $PINTOS_HOME/src/threads
ii) make
iii) make check

These three steps will compile the code (step ii) and then run all of the tests for project 1. If this goes well, it means your machine is properly configured for you to start coding for project 1.

NOTE:-

Following link is also useful as far as installation is concerned:-

https://tssurya.wordpress.com/2014/08/16/installing-pintos-on-your-machine/

Tasks

Create a write-up describing what you have done so far with respect to installing and running Pintos, and what you have understood about the assignment and about the starter code.

Deliverables
Text Document containing necessary details.