ICS-OS Lab 01: Building and Booting ICS-OS

Objectives

At the end of this activity, you should be able to:

1. set up the build environment;
2. build the kernel binary image;
3. build the distribution floppy disk image; and
4. boot ICS-OS and run two commands.

1 Introduction

ICS-OS is an instructional operating system that can be used for understanding different operating system concepts. An operating system is no different from other software in that it is written in a programming language, such as C.

The build process creates the compressed kernel binary image (`vmdex`) and the floppy disk image (`ics-os-floppy.img`). Since ICS-OS has a relatively large number of source files, the `Make` utility is used for the build. You can examine the contents of `Makefile` to learn more of the details how this is done.

To start ICS-OS, the floppy disk image is set as the boot device in Qemu, an emulator for various machine microarchitectures. The floppy disk image contains the GRUB bootloader which transfers control to the ICS-OS kernel (`vmdex`). After the boot process, a prompt is provided for users to enter commands.

2 Prerequisites

The recommended development environment is Ubuntu 20.04 or later. The actual building of the kernel is done inside an Ubuntu 16.04 container. Familiarity with the command line is also needeed. GitHub Codespaces can also be used but will not have graphics support. Install the following packages on your local machine/VM or in Codespaces terminal after opening the repo.

```
$sudo apt update
$sudo apt install qemu-system-x86 gcc-multilib git
```

3 Deliverables and Credit

Perform the tasks below and capture screenshots while you do them. Submit a PDF file containing the screenshots with captions. Do not forget to put your name and laboratory section. Credit is five (5) points.

4 Tasks

Task 1: Install Docker and Docker-Compose

`docker`[1] and `docker-compose`[2] should be installed to build in the latest versions of Ubuntu (see footnote for links to the guides). Also perform the post-install steps[3].

In Codespaces, these are already installed.

[1] https://docs.docker.com/engine/install/ubuntu/
[2] https://docs.docker.com/compose/install/linux/#install-using-the-repository
[3] https://docs.docker.com/engine/install/linux-postinstall/

Task 2: Clone the repository and explore the source tree

ICS-OS[4] is open source and is hosted on Github. Run the following command to checkout the source code and explore the source tree. This step should be done if you are not using Codespaces.

```
$git clone https://github.com/srg-ics-uplb/ics-os.git ics-os-yourinitials
```

In the command above, the source will be cloned in the `ics-os-yourinitials` directory. We will refer to this directory in this document as `$ICSOS_HOME`. You can also use the `export` command to define it as an environment variable.

We create a branch for lab01 to make code management easier.

```
$cd $ICSOS_HOME
$git checkout -b lab01
$git branch  #check that you are in the lab01 branch
```

Task 3: Build ICS-OS kernel

The build process must be done inside a docker container(an Ubuntu 16.04 environment). There is a `$ICSOS_HOME/ics-os/docker-compose.yml` and a `$ICSOS_HOME/ics-os/Dockerfile` that describe the build environment.

Open a new terminal. Start and enter the container using the commands below.

```
$cd $ICSOS_HOME/ics-os
$docker compose run ics-os-build
```

You will be dropped to a root shell inside the container where you can perform the build. The `$ICSOS_HOME/ics-os` directory is mapped to `/home/ics-os` inside the container. Thus, you can perform the edits outside the container(in another terminal) and the changes will be reflected inside the build environment. The following steps will build the kernel image.

```
:/#cd /home/ics-os
:/#make clean
:/#make
```

Note that the details of the steps are in the `$ICSOS_HOME/ics-os/Makefile`.

Task 4: Create the disk and boot ICS-OS

Open a new terminal. Build the boot floppy then start Qemu with the floppy image as boot device using the commands below.

```
$cd $ICSOS_HOME/ics-os
$sudo make floppy
$make boot-floppy
```

[4] https://github.com/srg-ics-uplb/ics-os/

In GitHub Codespaces, the following commands must be used since its harder to set up the GUI.

```
$sudo make floppy
$qemu-system-i386 -fda ics-os-floppy.img -curses
```

You should now see the GRUB boot menu. Simply press enter to boot ICS-OS. Note that the details of the steps are in the `$ICSOS_HOME/ics-os/Makefile`.

Task 5: Run ICS-OS commands

Once the ICS-OS command prompt (`%`) appears, type `help`. Examine the list of commands and run two of these commands. Do not forget to capture screenshots of the outputs.

Task 6: Cleanup

To exit the build container.

```
:/#exit
```

Go back to the master branch of the source code.

```
$git checkout master
$git branch
```

5 Reflection

Write some realizations and questions that crossed your mind while doing this lab.

