

COMP2013 Systems Engineering

I

Group Project First Proof of Concept Demonstrator

Due by 11.55pm Wednesday 9th January 2013

Submission via Moodle and Version Control

Repository

Overview

The goal is to create a Proof of Concept (PoC) implementation of an application specified by a client, to run within a custom operating system (OS) environment.

You must form into teams of 4-5, and each team will bid for first, second and third choice projects. Once bidding is complete, teams will be informed of which project they have been awarded. The summaries for the projects available to bid for are available on Moodle.

The projects will continue through COMP2014 in term 2, with the final deliverable being a complete, tested, deployable application.

Learning Outcomes

The specific learning outcomes for COMP2013 are:

- To investigate practical solutions to a challenging real world application development problem.
- To learn about, and make use of, a range of commercial and open source tools and technologies.
- To work with a substantial source code base and use a shared version control system such as Git to hold the project artefacts.
- To configure a virtual machine and/or OS for development work and install a suitable tool chain, including an IDE, code management tools and code libraries.
- To start learning about testing and the use of testing tools.

- To investigate the customisation of an operating system to support the deployment and running of a custom application.
- To function as a team member within a group development project.
- To manage a development project effectively.

Background

Last year in COMP1010 you had to write programs that made use of a function call API to get robots to perform tasks such as mapping and running races. That API enabled the C programs you wrote to make use of the hardware features on the robots, by reading sensor measurements, accessing the motor encoders and infrared emitters, and so on. You also made use of a tool chain for cross compiling code, allowing programs to be tested on X86 processors and then deployed on ARM processors (on the mobile phone device). The robot hardware and the API provided a platform on which you could experiment and essentially write proof of concept programs to demonstrate that the robots could be made to behave in interesting ways.

Consumer products such as set top boxes, smart TVs, mp3 players, kiosks, tablet and phone devices are first constructed as proof of concept (PoC) demonstrators by creating a series of prototypes to explore a range of possible solutions. One of the fastest growing industry methods for developing prototype APIs on new hardware is to use a well-established component based OS implementation, such as Debian GNU/Linux, stripped down to the essential components only and running on minimum spec hardware. The aim is to identify a baseline specification to determine what features are required to support the device or application, along with the design and cost benefits and trade-offs – good engineering practice!

With a feasible specification in place, the final combination of required services and capabilities can be selected for the desired product. Also additional hardware choices can be explored based on costs, technology innovation and timeframe for development.

Modern systems engineering processes will prototype and proof the production of a new platform through:

- Creating a tool chain and processes for compiling and linking components and libraries for a target platform (ARM/X86/MIPS etc.).
- Selection and integration of existing components and APIs such as network stacks, graphics devices, human interface device drivers, codecs and so on.
- The development of a series of abstraction layers for higher level software to make use of the hardware, avoiding the need for application developers to access the low level layers or hardware directly.

- Developing test suites and performance benchmarking.

The Project Work

Starting in the second half of COMP2013 and continuing throughout COMP2014 in term2 you will work through a system design, build and test cycle for an application for a real client. In COMP2013 you will be focusing on building your first proof of concept demonstrator for the requirements given. In COMP2014 you will transform the demonstrator into a working solution and develop the software abstraction layers that will allow the creation of applications and services for the platform you are working on. For both COMP2013 and COMP2014 you will be coordinating and publishing your work via an open source version control repository such as GitHub.

The key PoC objectives are:

- Create a team version control repository for your project, which all team members will use and will contain all the source code artefacts and documentation. The repository must be used properly, so that all changes are checked in promptly creating a clear record of the development process.
- Research and investigate one or more Linux OS distributions for relevant features such lightweight capabilities, ability to customise for your project and driver compatibility.
- Explore ways to configure an OS in a virtual machine, or on a real device, to determine how the OS can be customised and streamlined to suit the project you are working on. There will be hardware devices available to work with (e.g., PCs and tablets).
- Configuration of launch and shutdown scripts for the target user domain – there should be no visibility of the underlying OS to client users.
- Remove unwanted packages and dependencies from the OS – the aim is to reduce OS footprint and to streamline the selection of necessary components for the target domain requirements. You should seek out a the minimum but optimal settings, exploring alternatives and trade-offs. There should be evidence of benchmarks and testing for minimum RAM/Processor/Graphics/Network capabilities for appropriate quality of service delivery.
- Test with virtual machines and report on testing data using test tools and frameworks in order to build a first-cut prototype of the application.

Team Management

Each group will have a team manager who will be assigned randomly and will hold the responsibility for delegation of work to the team. This module requires

you to show clear evidence of your individual contribution to teamwork, and assessment will take account of this.

A secondary team lead will have responsibility for the reporting across the project tasks. Each member of the team will take a primary and a secondary role to ensure that fair responsibilities are shared. Teams will use the labs and arrange meetings outside of lab time with minutes taken of each meeting (including who is present); you should identify and use software tools for sharing and distributing workload e.g., Google docs, or cloud-based project management tools.

Project Deliverables

Main Report (50 marks)

Submit by 11.55pm 9th January 2013 via upload on Moodle, as a single pdf document, one copy per group. The report should not exceed 50 pages (sides of A4) in length. The report content should cover the following:

1. Research and investigation for the PoC Demonstrator (20 marks)
 - Identification of a suitable OS, by comparing the alternatives (e.g., Linux distributions) and selecting the one with the most appropriate features.
 - Identification and trial of appropriate optional packages suited for the target platform.
 - Development of the tool chain to enable building for the target, including a minimum hardware specification.
 - Development of configuration scripts, version control method, such as GitHub or other open source project platform and installation process.
 - Critical discussion on performance and capabilities for the target hardware.
2. System Components Integration (15 marks)
 - Component selection from the optional packages available.
 - Discussion on the integration process.
 - Discussion on the removal of any unnecessary OS components.
 - Scripts for default launch behaviour.
 - Component map, identifying the layers for abstraction needed for development of applications (e.g., installation of a Web Server component).
 - Identification of what has been adapted from the open source community and what is to be newly developed or configured.

3. Test plan (15 marks)

- Investigation of relevant available testing tools.
- Development of a test suite including unit tests, virtual machine tests, component integration tests, quality of service tests and benchmarking.
- Identification of the test tools that would be required for the apps layer in COMP2014.
- Report on performance charts for specific VM configurations, including minimum and optimal hardware.

Video Report (10 marks)

Submit by 11.55pm 9th January 2013 via upload to YouTube, one video per group. The YouTube URL should be submitted via Moodle. The video does not have to be made public on YouTube, as long as it is accessible to the project markers. The maximum length is 15 minutes (default YouTube limit). If possible submit in 720p format, so that screen details are visible.

The video should:

- Present the first Proof of Concept OS design, with an overview of the features available and a demonstration of the OS in action, including start-up and shut-down.
- Describe the key requirements, the decisions made on how to customise and streamline the OS
- The tools used and the tool chain provided, and any other relevant features.

Individual report (25 marks)

Submit by 11.55pm 9th January 2013 via upload on Moodle, as a single pdf document. The report should not exceed 10 pages (sides of A4) in length.

The report should provide:

- A summary of personal achievements made on the project, and your contributions and decisions made that affected the development of the PoC.
- A critical assessment of each team members contribution, including your own, in their primary and secondary roles.
- Your personal assessment of the system architecture, quality of the decisions made, whether the system is fit for purpose and how the further development of the system should proceed.

