Adapting the component platform CAmkES to Linux

Author: Hanzhang Zeng
Supervisors: Dr. Ihor Kuz, Matthew Fernandez, Siwei Zhuang
Research Theme: The Digital Future

Facts
1. A component platform helps software development
   • It is a tool to architect software systems
   • It constructs application as a set of components and connectors
   • Each component is highly reusable
2. CAmkES is a component platform running on seL4
   • It stands for Component Architecture for microkernel-based Embedded Systems
   • It generates glue code for connectors between components of applications
   • It supports applications running on seL4 kernel, a high assurance microkernel
3. Linux has lots to offer
   • Free device drivers
   • Powerful debug toolchains

Aim: CAmkES on Linux
• Add Linux platform support for CAmkES applications.
• Adapt CAmkES build system to Linux.

Methodology

Current situations
1. CAmkES generated code only works on seL4: CAmkES generates C glue code from architecture descriptions and combines it with the user-defined source code to compile seL4 binary files.
2. The build system is seL4 specific: CAmkES build system takes in seL4 configurations and groups binary files into a bootable image for development boards.

Solutions
1. Adapting CAmkES object implementations to Linux:
   In order to support the Linux platform, seL4 kernel objects are replaced with Linux IPC facilities as you can see in Fig. 2. System calls in glue code templates are changed from seL4 to Linux accordingly.
2. Simplifying CAmkES build system:
   CAmkES build system on Linux does not require seL4 related configurations and bootable image generation. Therefore, the build system is refined and it only provides basic functionalities for compiling source code into binary files.

Results
1. Successfully ported CAmkES objects to Linux:
   Glue code templates of component, connectors and utilities like lock and semaphore are adapted to Linux.
2. Successfully adapted build system to Linux:
   The build system renders glue code, compiles binary files and links them with Linux libraries properly.

Impact
1. A richer environment for testing embedded system
2. A boost for embedded system development
3. A demonstration of CAmkES platform independency

Future Work
• Support thread priority and scheduling configuration
• Polish the build system and make it more user-friendly
• Support device driver programming on Linux

Fig. 1: User space device drivers on seL4 and kernel space drivers on Linux

Fig. 2: Table of equivalent implementations

Fig. 3: CAmkES applies templates according to the connector type during glue code generation

Fig. 4: Reusable component (Client) and platform specific component (Timer) generates binary files for different platforms