Embedded Systems Programming
Virtualization

Iwona Kochanska

Department of Marine Electronic Systems
Virtualization = abstraction of resources. Multiple operating systems on the same hardware and OS at maximum possible efficiency. A guest operating system processes (emulated system) are performed directly on the hardware of host computer. Only in situations where such operations can’t run directly, virtualizer emulates them.

- **API emulation** - introduction to the main OS the API from another system. Example: Wine (Wine is not an emulator) - implementation of environmental WinAPI for Unix / X11.
Virtualization

▶ **Full emulation** – way to run applications from incompatible computer in relation to that used (eg. PC / Mac).
  ▶ basic components (CPU, RAM, HDD, CD, etc.) and operating system (virtual OS) are emulated, providing high portability
  ▶ emulator performs in the loop all that would do the actual processor of emulated machine, which leads to a drop in performance of host computer. Example: QEMU.
Virtualization

**Hypervisor** - Virtual Machine Manager. Tool for managing the virtualization process.

- Decide which processes of the virtualized OS can be performed directly on the hardware and which should be emulate.
- If the specified operation can not be performed directly on the hardware resource, it is captured and emulated by the hypervisor.
- Mediates between the virtual interrupt system and hardware resources
- Examples: *VMware Workstation*, *Virtual Box*
Virtualization
Docker - containers

Containers - a way to separate applications from the operating system and the physical infrastructure used to connect to the network.

- Installed outside the operating system kernel and virtualizes the specific application environment.
- Archetype - chroot in UNIX systems, tool for limiting the resources that can be utilized by the process and its children.
- Containers compatible with Linux (LXC, Linux Kernel Containers) - one of the most modern methods of application virtualization
  - allocate CPU, memory, harddrive and network resources for applications isolated from the OS
  - separate process-trees, network access, user ID, access to files
  - are flexible as they allow an administrator to virtualize a single application, not the entire operating system using the VM.
Docker - LXC-based application

- Manages images and assists in the implementation of application virtualization.
- Provides automation and mechanisms for rapid creation of containers LXC.
- Internal project of dotCloud company. Public since March 2013.
- Integrated with a number of other tools, among others, Ansible, Chef, OpenStack, Puppet, Salt. It is also included with RHEL, OpenShift PaaS Google Compute Engine, Deis, as well as Amazon Web Services Elastic Beanstalk.
- A standard of application virtualization for Linux.
Docker vs Virtual Machine

Virtual Machines

- Hypervisor
- Host OS
- Server
- MySQL
- Bins/Libs
- Guest OS

Containers

- Container Engine
- Host OS
- Server
- MySQL
- Bins/Libs

Source: http://patg.net/containers,virtualization,docker/2014/06/05/docker-intro/
Docker

- Homepage:
  https://www.docker.com/

- Installation in Debian OS:
  $ sudo apt-get install docker.io

- First run (creating the container):
  $ sudo docker run image_name

- Run the container:
  $ sudo docker start container_name

- Run with shared directory:
  $ sudo docker run --v /var/logs/on/host:/var/logs/in/container
QEMU emulator

Quick EMUlator - fast emulator available as open source software.

- Allows to run multiple operating systems simultaneously on one machine.
- Emulates multiple CPU architectures.
- Can operate in two modes:
  - user mode – to run Linux processes compiled on another type of processor (eg. 64-bit applications on 32-bit processor).
  - systemu – emulates the entire system, including the processor, hard drive and the corresponding peripheral devices. Operating system can be run and installed on different architectures: x86, x86_64, ARM, SPARC, SPARC64, PowerPC, PowerPC64, MIPS, m68k (Coldfire), SH-4, Alpha, CRIS.
References

http://patg.net/containers,virtualization,docker/2014/06/05/docker-intro/