

CGroups y Namespaces

Explicación de práctica 4

Sistemas Operativos

Facultad de Informática
Universidad Nacional de La Plata

2025



Repositorio con código para la práctica



<https://gitlab.com/unlp-so/codigo-para-practicas>

Virtualización y emulación



Algunos motivos para virtualizar / emular

- Ejecutar software hecho para otro:
 - Sistema operativo
 - Arquitectura
 - Ambos
- Ejecutar múltiples instancias de distintos sistemas operativos
- Ejecutar múltiples instancias del mismo sistema operativo
- Ejecutar una aplicación de forma aislada por:
 - Seguridad
 - Facilidad de instalación (dependencias, bibliotecas, aplicaciones legacy)



Emulación



Emulación

- QEMU
 - Full System Emulation
 - Alpha - Legacy 64 bit RISC ISA developed by DEC
 - Arm (arm, aarch64) - Wide range of features, see A-profile CPU architecture support for details
 - AVR - 8 bit micro controller, often used in maker projects
 - Hexagon - Family of DSPs by Qualcomm
 - PA-RISC (hppa) - A legacy RISC system used in HP's old minicomputers
 - x86 (i386, x86_64) - The ubiquitous desktop PC CPU architecture, 32 and 64 bit.
 - LoongArch - A MIPS-like 64bit RISC architecture developed in China
 - m68k - Motorola 68000 variants and ColdFire
 - Microblaze - RISC based soft-core by Xilinx
 - MIPS (mips*) - Venerable RISC architecture originally out of Stanford University
 - OpenRISC - Open source RISC architecture developed by the OpenRISC community
 - Power (ppc, ppc64) - A general purpose RISC architecture now managed by IBM
 - RISC-V - An open standard RISC ISA maintained by RISC-V International
 - RX - A 32 bit micro controller developed by Renesas
 - s390x - A 64 bit CPU found in IBM's System Z mainframes
 - sh4 - A 32 bit RISC embedded CPU developed by Hitachi
 - SPARC (sparc, sparc64) - A RISC ISA originally developed by Sun Microsystems
 - Tricore - A 32 bit RISC/uController/DSP developed by Infineon
 - Xtensa - A configurable 32 bit soft core now owned by Cadence

QEMU otros modos

- QEMU User space emulator
 - Traduce las instrucciones de un programa.
 - No emula todo el hardware, no carga un kernel en el entorno emulado, solo programas.
 - Mismo OS pero distintas arquitecturas.
 - Ante una system call traduce la system call (convirtiendo endianness si es necesario) e invoca a la system call equivalente.
 - Solo Linux y BSD.



QEMU system ejemplo

Emularemos hardware de Sun Microsystems de los años 90

- SPARCstation 5
- CPU SPARC (RISC, 32bits, big endiann)
- Disco SCSI
- 32 MiB de RAM
- Placa de Red [Ethernet AMD Lance](#) 10mbps



Ejecutaremos SunOS 4 (Solaris 1) sobre ese hardware emulado



Virtualización



Virtualizadores tipo 1 y 2

- Tipo 1
 - Bare metal (se ejecutan directo sobre el hardware)
 - El Hypervisor accede al hardware directamente
 - Ejemplos: Xen, VMWare ESXi, KVM (integrado en el kernel Linux)
- Tipo 2
 - El Hypervisor se ejecuta sobre un sistema operativo
 - El Hypervisor accede al hardware a través del SO Host
 - Ejemplos: Virtualbox, VMWare Workstation.



QEMU System Emulation + Accelerator

- Permite virtualizar usando hypervisors
 - KVM - Linux
 - Xen - Linux (as dom0)
 - Hypervisor Framework (hvf) - MacOS
 - Windows Hypervisor Platform (whpx)
 - otros...



chroot



chroot

- Hace creer a un proceso que / es una carpeta
- Permite ejecutar procesos de forma “aislada”
- Los procesos pueden tener sus propias versiones de bibliotecas
- Podríamos ejecutar programas de otra distribución
- Se comparte el mismo kernel
- DEMO



CGroups



CGroups

- Resource Limiting: grupos no pueden excederse en la utilización de un recurso (tiempo de CPU, cantidad de CPUs, cantidad de memoria, I/O, etc.)
- Prioritization: un grupo puede obtener prioridad en el uso de los recursos (tiempo de CPU, I/O, etc.)
- Accounting: permite medir el uso de determinados recursos por parte de un grupo (estadísticas, monitoreo, billing, etc.)
- Control: permite freezar y reiniciar un grupo de procesos



CGroups

- Resource Limiting: grupos no pueden excederse en la utilización de un recurso (tiempo de CPU, cantidad de CPUs, cantidad de memoria, I/O, etc.)
- Prioritization: un grupo puede obtener prioridad en el uso de los recursos (tiempo de CPU, I/O, etc.)
- Accounting: permite medir el uso de determinados recursos por parte de un grupo (estadísticas, monitoreo, billing, etc.)
- Control: permite freezar y reiniciar un grupo de procesos



CGroups

- Configuraremos los CGroups usando `/sys/fs/cgroups`
- Existen otras formas:
 - `cgexec`
 - `systemd` (se pueden configurar cgroups para los servicios)
- DEMO



Namespaces



Namespaces

- Permiten aislar lo que los procesos ven
- Se puede asignar un proceso al crearlo (`clone()` o `unshare()`)
- Se puede asignar un proceso en ejecución (`setns()`)



Namespaces

Entre los namespaces provistos por Linux:

- IPC: Flag: CLONE_NEWIPC. System V IPC, cola de mensaje POSIX
- Network: Flag: CLONE_NEWNET. Dispositivos de red, pilas, puertos, etc
- Mount: Flag: CLONE_NEWNS. Puntos de montaje
- PID: Flag: CLONE_NEWPID. IDs de procesos
- User: Flag: CLONE_NEWUSER. IDs de usuarios y grupos
- UTS: Flag: CLONE_NEWUTS. Hostname y nombre de dominio
- Cgroup: Flag: CLONE_NEWCGROUP. cgroup root directory
- Time: Flag: CLONE_NEWTIME. Distintos offsets al clock del sistema por namespace



Namespaces

- DEMO



¿Preguntas?

