Microkernel Tidbits

http://d3s.mff.cuni.cz/aosy
Genode

Operating system component framework

Genode/seL4 Demo

```bash
git clone -b sel4_screenshot https://github.com/alex-ab/genode.git
git clone https://github.com/genodelabs/genode.git
cd genode/

./tool/create_builddir sel4_x86_32 BUILD_DIR=genode.sel4
vi genode.sel4/etc/build.conf

wget http://ftp.fau.de/qtproject/archive/qt/5.1/5.1.0/single/qt-
everywhere-opensource-src-5.1.0.tar.gz
mkdir contrib/qt5-8c8f232dca82e015b4f25f4fa99e7c7eeecc128.incomplete
mv qt-everywhere-opensource-src-5.1.0.tar.gz \
    contrib/qt5-8c8f232dca82e015b4f25f4fa99e7c7eeecc128.incomplete/

./tool/ports/prepare_port qt5 dde_linux ffat freetype icu jpeg libc \
    libpng lwip mesa sel4 stdcxx x86emu zlib qoost coreutils ncurses \
    vim bash diffutils less

make -C genode.sel4 run/noux
make -C genode.sel4 run/noux_bash
make -C genode.sel4 run/sel4
```
Microkernel provides capability mechanisms

User space is responsible for managing the capability space by deriving capabilities from the untyped memory capability.
Operations with Capabilities

- **Cloning**
  - Creating a new capability pointing to the same resource as the original capability

- **Minting**
  - Creating a new capability with a subset of permissions to the original resource

- **Deriving**
  - Creating a new capability based on a hierarchy of capability types
    - Untyped memory (representing a region of physical memory)
      - Can be split into several untyped memory regions
      - Can be retyped to a different capability (thus creating a different type of kernel object)
seL4 Capabilities

**Kernel objects**

- UntypedObject (physical memory range)
- TCBObject (thread)
- EndpointObject (IPC calls destination)
- AsyncEndpointObject (signal recipient)
- CapTableObject (array of capabilities)
- X86_4K (4 KB frame)
- X86_4M (4 MB frame)
- X86_PageTableObject (2nd level page table)
- X86_PageDirectoryObject (1st level page table)
seL4 Capabilities Memory Management

seL4_X86_UnTyped_Retype(cnode_selector(phys_addr),
    seL4_X86_4K, ..., ..., ..., ..., ..., phys_addr >> FRAME_WIDTH, 1);

seL4_X86_UnTyped_Retype(cnode_selector(pt_phys_addr),
    seL4_X86_PageTableObject, ..., ..., ..., ..., ...
    pt_phys_addr >> FRAME_WIDTH, 1);

seL4_X86_UnTyped_Retype(cnode_selector(pd_phys_addr),
    seL4_X86_PageDirectoryObject, ..., ..., ..., ..., ...
    pd_phys_addr >> FRAME_WIDTH, 1);

seL4_X86_PageTable_Map(cnode_selector(pt_phys_addr),
    cnode_selector(pdphys_addr), virt_addr,
    seL4_X86_Default_VMAttributes);

seL4_X86_Page_Map(cnode_selector(phys_addr),
    cnode_selector(pd_phys_addr), virt_addr, seL4_AllRights,
    seL4_X86_Default_VMAttributes);
Genode Resource Management

application-specific TCB

explicit resources assignment

resources attached to sessions

Hypervisor (Type 1)

- **Hypervisor (Type 1)**
- **Operating System**
  - Kernel
  - Memory mgmt
  - Scheduler
  - Communicator
- **App**
- **Unprivileged Mode**
- **Privileged Mode**
- **Hardware**
Microhypervisor

- Network stack
- Security server
- Device multiplexer
- File system multiplexer
- Device driver server
- File system driver server
- ... (unprivileged mode)
- Memory management
- Scheduler
- IPC
- Microhypervisor
- Hardware

Hyper-privileged mode
NOVA Microhypervisor

NOVA Microhypervisor

- x86-64
  - Multiprocessor support
  - Mandatory Intel VT-x and optional Intel VT-d (IOMMU) for isolation
  - IPC derived from traditional L4
    - Synchronous, rendez-vous style
    - Capability-based
  - Support for running unmodified guest OSes via a VMM
    - VirtualBox in Genode/NOVA
Separation Kernel

- **Hypervisor eliminating covert channels**
  - Concept by John Rushby (1981)
    - Creating an environment indistinguishable from a physically distributed system
    - Each protection domain acts as a separate, isolated machine
    - Information can only flow from one machine to another along known external communication lines
  - Supporting mixed-criticality protection domains
  - INTEGRITY-178B
    - First separation kernel certified by NSA for safety-critical applications (e.g. aircraft avionics consolidation)
Muen Separation Kernel

\textbf{x86-64}

- Written in SPARK 2014 (2500 lines + 300 lines of assembler)
  - Formally defined subset of Ada with contract specification
- Multiprocessor support
- Mandatory both Intel VT-x and Intel VT-d (IOMMU) for isolation
  - Fixed cyclic scheduling
  - Static assignment of resources

\textbf{Subjects}

- Native 64bit Ada / SPARK 2014
- Genode/base-hw
  - GNU/Linux, Windows VM via VirtualBox in Genode

\textbf{IPC}

- Shared memory buffers
- Virtual interrupts (events)
User Space Pagers in HelenOS

HelenOS memory management

- Traditional kernel architecture (frame allocator, kernel heap allocator, virtual address space areas manager)
  - Mostly mechanisms, but also unavoidable minimal policies (single point of failure)
  - Originally three address space area backends
    - backend_phys
    - backend_anon
    - backend_elf
  - New address space area backend
    - backend_user
User Space Pagers in HelenOS
User Space Pagers in HelenOS

as_area_create(..., pager_info)

IPC connection

memory object
User Space Pagers in HelenOS

IPC_M_PAGE_IN(offset, memory object)

Pagee

Pager

Kernel

blocked in page fault handler
User Space Pagers in HelenOS

Diagram:
- Pagee
- Pager
- Kernel

ipc_answer_1(EOK, virtual address)

Wake up and map page to physical address

find physical address
User Space Pagers in HelenOS

- Pagee
- Pager
- Kernel
Q&A