- **Mike Dodson** • 16 days ago

  What's the difference between a `cptr` member of a `vka_object_t` (e.g., `tcb_object`) and a `capPtr` member of a `cspacepath_t` (such as `ep_cap_path` in exercise-6?

  - Reply
  - Share ›

Ammar S Salman Mike Dodson • 16 days ago

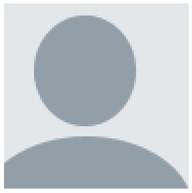
They are all capability pointers pointing to different capabilities. For the child process, we used ep_object.cptr to make a cspace path which we used to mint the endpoint to the child process. This allows both main and child processes to get pointer to the endpoint capability responsible for their communication (used by both for communication).

On the other hand, the tcb_object.cptr is a capability pointer to the child thread's TCB, which allows main thread to control execution of child thread. In the exercise, we used it to set execution priority, and we also used it to start (more accurately, resume) the child thread.

While they all are capability pointers, they represent different types of capabilities.

Finally, vka_object_t represents the generic object type for vka objects. It can be a TCB object, an endpoint, ..., etc.

- Reply
- Share ›



Mike Dodson Ammar S Salman • 16 days ago

But does `capPtr` mean something fundamentally different from `cptr`, or are they both capability pointers and the structures just have different member names?

- •
- Reply
- •
- Share ›

Ammar S Salman Mike Dodson • 16 days ago • edited

They are both capability pointers, fundamentally the same but just different member names.

Edit: just for further confirmation, you can find the definition of cptr as part of vka_object in:

```
projects/seL4_libs/libsel4vka/include/vka/object.h
```

And definition of capPtr in:

```
projects/seL4_libs/libsel4vka/include/vka/cspacepath_t.h
```

You will notice they are defined as seL4CPtr:

```
typedef struct vka_object {
seL4_CPtr cptr;
seL4_Word ut;
seL4_Word type;
seL4_Word size_bits;
} vka_object_t;

typedef struct _cspacepath_t {
seL4_CPtr capPtr;
seL4_Word capDepth;
seL4_CNode root;
seL4_Word dest;
seL4_Word destDepth;
seL4_Word offset;
seL4_Word window;
} cspacepath_t;
```

- Reply
- Share ›

Mike Dodson Ammar S Salman • 16 days ago

Got it. Thanks. That's very helpful (I should have just dug into the structures myself...).

- 1
- Reply
- Share ›

Mike Dodson • 16 days ago • edited

Any advice on navigating documentation? I found the seL4 lib API documentation, but I can't find similar documentation for sel4utils. I'm trying to understand calls like `sel4utils_set_instruction_pointer`.
Atm, I'm just googling and finding descriptions in the online tutorials.

- Reply
- Share ›

Fabrizio Bertocci Mike Dodson • 16 days ago

One thing that I normally do is use ctags to parse the directories containing source and headers. From an example directory, run:

```
ctags -R --exclude *.tex ../kernel ../projects src
```

(you need to tell ctags to exclude all the tex files otherwise it will end up producing a corrupted tag file)

Then if you use vim, open your source code:

```
vim src/main.c
```

from inside vim, then when you press CTRL+] it'll go to the definition of the function under the cursor.

If there are multiple symbols with the same name, press g, then CTLR+].

- 1
- •
- Reply
- •
- Share ›

Mike Dodson Fabrizio Bertocci • 16 days ago • edited

Brilliant. Works perfectly (needed to add an equal sign in the exclude option).

```
ctags -R --exclude=*.tex ../kernel ../projects src
```

Thanks!

- •
- Reply
- •
- Share ›

Fabrizio Bertocci Mike Dodson • 15 days ago

Oops, right! Thanks for checking.

- Reply
- Share ›

Ammar S Salman Mike Dodson • 16 days ago

That's one of the hard things to navigate as they do not have online documentation, nor offline frankly. During our development we had to look through the header files under include directories to find what functions we could use.

You may find the following command useful:

```
grep -IR sel4utils_set_instruction_pointer --include *\.h
```

This helps you find the function. This function is architecture-specific and therefore it's in sel4_arch_include. Even still, the function you specified is not documented in the header file.

For completeness, the function we used comes from:

```
projects/seL4_libs/libsel4utils/sel4_arch_include/x86_64/sel4utils/sel4_arch/util.h
```

- Reply
- Share ›

Mike Dodson Ammar S Salman • 16 days ago

Thanks for the quick response. That's helpful. I appreciate it.

- •
- Reply
- •
- Share ›

Ammar S Salman Mike Dodson • 16 days ago

You're welcome!

- •
- Reply
- •
- Share ›

Jacob Saina • 16 days ago • edited

The VM does not load for me in VirtualBox. I get:

```
Failed to import appliance .../seL4-IAI.ova.

Error reading OVA '.../seL4-IAI.ova' (VERR_TAR_UNEXPECTED_EOS).

Result Code: VBOX_E_IPRT_ERROR (0x80BB0005)
Component: ApplianceWrap
```

```
Interface: IAppliance {8398f026-4add-4474-5bc3-2f9f2140b23e}
```

I downloaded the OVA from the Google drive link (twice, to verify that they matched and it wasn't a download error).

- •
- Reply
- •
- Share ›

Ammar S Salman Jacob Saina • 16 days ago

You may want to try:
1. Updating VirtualBox.
2. Verify the filesystem is NTFS (under Windows) or ext4 (under Linux).
3. If the problem persists, the file may be corrupted. But given you have downloaded it multiple times to verify integrity that seems a bit unlikely. We have a 2 other backup links, they are password protected and can be found in the seL4 summit email information.

- •
- Reply
- •
- Share ›

Ammar S Salman Jacob Saina • 16 days ago • edited

What file system are you running under? File systems like FAT can't handle large files like the VM in some cases.

Also, what version of VirtualBox are you using?

- •

- Reply
- •
- Share ›

Jacob Saina Ammar S Salman • 16 days ago

I was on 5.something, and upgraded to the current version. I can open the VM now.

- •
- Reply
- •
- Share ›

Ammar S Salman Jacob Saina • 16 days ago

Expected. VirtualBox 5-5.2 have had similar issues. We've done our testing on more recent versions.

- •
- Reply
- •
- Share ›

Renato Levy • 16 days ago

We have tried to dimension this self pace training to be doable in a couple of hours. As you finish it, we would love some comments and the feedback on how it took for you to complete.

- 1
- •
- Reply
- •
- Share ›

Jacob Saina Renato Levy • 15 days ago

I spent about 4 hours, got partway through exercise 5, and stopped.

- •
- Reply
- •
- Share ›

Mike Dodson Renato Levy • 16 days ago

It took me about 4 hours. I found the first half very helpful. The second half was interesting, but concepts were coming pretty quick, so I was completing the tasks without necessarily digging into everything that was happening. I'll come back to it later, though, and work through it slower. Thanks very much!

- •
- Reply
- •
- Share ›

Jeremy • 16 days ago

Hello! I'm trying to follow along with the tutorials and didn't have much luck with the VM. It seems some dependencies are missing jinja2 specifically. I tried installing via pip which I manually installed but that didn't work. Do you know if there is a specific set of pre-requisites we can install for this to get the provided VM ready to go?

- •
- Reply
- •
- Share ›

Renato Levy Jeremy • 16 days ago

Were you able to get it to work?

- •
- Reply
- •
- Share ›

Jeremy Renato Levy • 16 days ago • edited

Yes, the docker piece works now. But the simulation throws a lot of errors and hangs.

SeaBIOS (version 1.12.0-1)

```
iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+1FF8FFA0+1FECFFA0
C980
Booting from ROM..Boot config: debug_port = 0x3f8
Boot config: disable_iommu = false
Detected 1 boot module(s):
module #0: start=0xa19000 end=0xa5dcc0 size=0x44cc0 name='images/hello-
world-'
Parsing GRUB physical memory map
Physical Memory Region from 0 size 9fc00 type 1
Physical Memory Region from 9fc00 size 400 type 2
Physical Memory Region from f0000 size 10000 type 2
Physical Memory Region from 100000 size 1fee0000 type 1
Adding physical memory region 0x100000-0x1ffe0000
Physical Memory Region from 1ffe0000 size 20000 type 2
Physical Memory Region from fffc0000 size 40000 type 2
Multiboot gave us no video information
ACPI: RSDP paddr=0xf58d0
ACPI: RSDP vaddr=0xf58d0
ACPI: RSDT paddr=0x1ffe156f
ACPI: RSDT vaddr=0x1ffe156f
Kernel loaded to: start=0x100000 end=0xa18000 size=0x918000 entry=0x10125e
ACPI: RSDT paddr=0x1ffe156f
ACPI: RSDT vaddr=0x1ffe156f
ACPI: FADT paddr=0x1ffe144b
ACPI: FADT vaddr=0x1ffe144b
ACPI: FADT flags=0x80a5
ACPI: MADT paddr=0x1ffe14bf
ACPI: MADT vaddr=0x1ffe14bf
ACPI: MADT apic_addr=0xfee00000
ACPI: MADT flags=0x1
ACPI: MADT_APIC apic_id=0x0
ACPI: MADT_IOAPIC ioapic_id=0 ioapic_addr=0xfec00000 gsib=0
ACPI: MADT_ISO bus=0 source=0 gsi=2 flags=0x0
ACPI: MADT_ISO bus=0 source=5 gsi=5 flags=0xd
ACPI: MADT_ISO bus=0 source=9 gsi=9 flags=0xd
ACPI: MADT_ISO bus=0 source=10 gsi=10 flags=0xd
ACPI: MADT_ISO bus=0 source=11 gsi=11 flags=0xd
ACPI: 1 CPU(s) detected
ELF-loading userland images from boot modules:
size=0x123000 v_entry=0x401107 v_start=0x400000 v_end=0x523000
p_start=0xa5e0000
```

```
Moving loaded userland images to final location: from=0xa5e000 to=0xa18000
size0
Starting node #0 with APIC ID 0
Mapping kernel window is done
Booting all finished, dropped to user space
hello world
Warning: using printf before serial is set up. This only works as your
printf is backed by seL4_Debug_PutChar()
hrintf is b
Caught cap fault in send phase at address 0
while trying to handle:
vm fault on data at address 0 with status 0x4
in thread 0xffffff801fe08400 "rootserver" at address 0
With stack:
0x41be98: 0x401740
0x41bea0: 0x0
0x41bea8: 0x401100
0x41beb0: 0x41bef0
0x41beb8: 0x40121b
0x41bec0: 0x0
0x41bec8: 0x41bf30
0x41bed0: 0x41bf20
0x41bed8: 0x41bf10
0x41bee0: 0x1
0x41bee8: 0x4010f2
0x41bef0: 0x41bff0
0x41bef8: 0x401394
0x41bf00: 0x0
0x41bf08: 0x524000
0x41bf10: 0x411018
```

see more

- Reply
- Share ›

Curtis Millar Jeremy • 16 days ago

There was a recent change made to sel4runtime that required the mechanism for process exit to be specified explicitly via `sel4runtime_set_exit`. If this isn't set, when `main` returns the initially configured callback (a null pointer) is dereferenced and that causes the fault.

The most simple way to resolve this can be seen in `sel4test` which suspends the initial thread when main returns.

- Reply
- Share ›

Ammar S Salman Curtis Millar • 16 days ago

Thank you for the clarification. Had we known this during development we would have set the function as done in the test repository.

- Reply
- Share ›

Jacob Saina Jeremy • 16 days ago

FYI I get (virtually) the same thing.

```
Booting all finished, dropped to user space
Hello, world!
Warning: using printf before serial is set up. This only works
as your
printf is backed by seL4_Debug_PutChar()
Hrintf is bac
Caught cap fault in send phase at address 0
while trying to handle:
vm fault on data at address 0 with status 0x4
in thread 0xffffff801fe08400 "rootserver" at address 0
With stack:
0x41be98: 0x401740
0x41bea0: 0x0
0x41bea8: 0x401100
0x41beb0: 0x41bef0
0x41beb8: 0x40121b
0x41bec0: 0x0
0x41bec8: 0x41bf30
0x41bed0: 0x41bf20
0x41bed8: 0x41bf10
0x41bee0: 0x1
0x41bee8: 0x4010f2
0x41bef0: 0x41bff0
0x41bef8: 0x401394
0x41bf00: 0x0
0x41bf08: 0x524000
0x41bf10: 0x41101c
```

see more

- Reply
- Share ›

Ammar S Salman Jacob Saina • 16 days ago

Yup, as expected. I have explained it in another reply.

Anyway, the reason printf prints all of these extra things is because we did not redefine the write function. We did not want to over-complicate hello-world and talk about redefining the write functions, especially given it's all meant for debugging purposes.

- Reply
- Share ›

This comment was deleted.

Jeremy Guest • 16 days ago

I added them to my above comment.

- Reply
- Share ›

Ammar S Salman Jeremy • 16 days ago • edited

Yes that is common, this isn't part of the program itself. We've done a lot of debugging and the error happens after the program returns and finishes execution.

We have done a lot of debugging regarding this issue and it happens with practically most applications. I suspect it could be related to the kernel. The issue was not present around 2 months ago and during development of this tutorial. But with more updates to seL4 repositories they started showing up.

Even running seL4-tutorials manifest (the one provided by seL4) yields the same error with hello-world and other tutorials.

We have found a few sources that suggest it is a bug in the kernel, and our debugging suggests the same. Again, the error occurs after the program executes 'return 0;'

For the purpose of this tutorial this should not be an issue.

- 1
- •
- Reply
- •
- Share ›

Ammar S Salman Jeremy • 16 days ago • edited

The dependencies are all within "container", make sure you launch it before you attempt building any exercises. Outside the container environment there are no installed dependencies because there would be a lot of duplication.

We recommend running "container" in exercises root directory, this should make it easier to navigate and use a single terminal for all exercises.

- 2
- •
- Reply
- •
- Share ›