The K Project

LSE Team

EPITA

April 26, 2017

User memory layout

Needed segments

- Code
- Data

Optional segments

- Stack

Sane memory layout

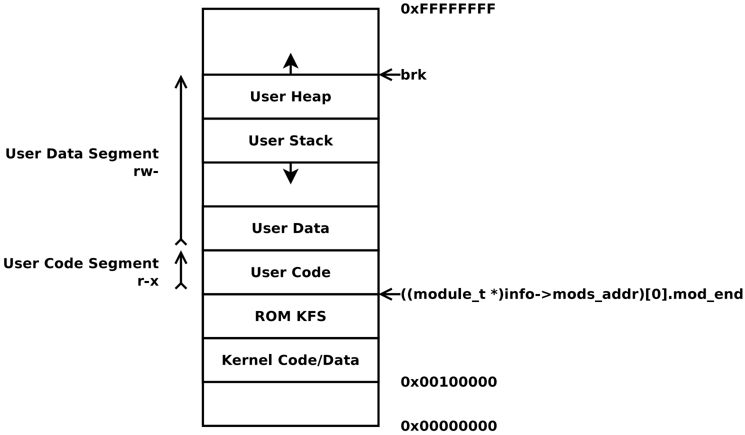

Figure: “Simple” example

What is needed

For every segments

- Find enough space with:
  - Multiboot's memory map
  - After the last module
- Should not overlap with each other

For the stack segment

- Should expand down

Sane memory layout (again)

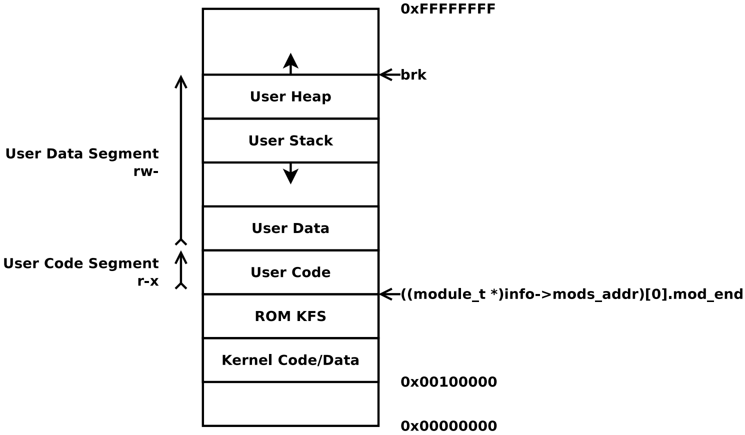

Figure: “Simple” example

Syscall

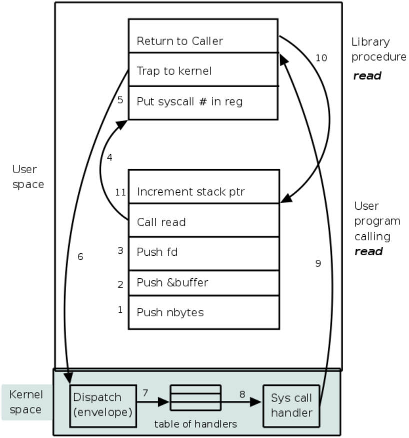

Figure: IA32 read syscall

Syscall Gate

- A unique syscall gate (0x80)
  - `int 0x80`
- `eax`: Syscall number
- `ebx`, `ecx`, `edx`: Syscall parameters

Implementation advices

- Jump table
- Do not forget to translate the user addresses
- Check for invalid user pointers

VGA Syscalls

setvideo

Swich between VGA text (3h) and graphic mode (13h)

swap_frontbuffer

Loads the user buffer into the graphic framebuffer

Syscall SBRK

Implementations advices

- `man 2 sbrk`
- Find some unused memory in the user data segment

Address space advices

- You can load and exec any ROM in “flat” mode.
- You can exec any ROM in kernel land
- GDB will not understand non-zero base address

Summary

- Implement the syscall handler
- Wrap and enable each syscall
- Implement the VGA syscalls
- Implement sbrk

Notes

All of these will be needed in order to run the ROMs.

Contact

- #k (irc.rezosup.org)
- labos.lse with [K] tag
- k[at]lse.epita.fr
- xdbob[at]lse.epita.fr
- pierre.marsais[at]lse.epita.fr