Lab 3 – Memory Management

202.1 Operating Systems

Objective

The objective of this laboratory is to understand how operating systems allocate memory by re-implementing a simplified version of Linux's `mmap` system call.

The estimated duration of this lab is **2 periods**. The result of Task 1 must be submitted on isc.hevs.ch/learn no later than Sunday 9th November at 23:59 (CEST).

Setup

You will need a modern C++ compiler to complete this lab. You can either install clang or gcc on your own system or use the instructions below to run a C++ compiler using Docker.

1. Install Docker Desktop and Visual Studio Code (VS Code) on your machine.
2. In VS Code, install the Dev Containers extension.
3. Download (or clone using git) the following repository: https://github.com/ISC-HEI/202.1-os.
4. Open the `lab-3` folder of this repository with VS Code and click on "Reopen in Container" in the bottom-right dialogue. If the dialog does not show, you can press `Ctrl+Shift+P` to open the command palette and type "Reopen in Container". VS Code will reopen and start a Docker container configured to run Linux on a 64-bit ARM architecture (emulated if necessary).

> Make sure to open the lab folder **directly** to let VS Code detect that a Docker container has been configured.

Whether or not you are using Docker, you can check if your system is ready by compiling the existing code with the following command:

```
make
```

Implementing `mmap`

Open the file `include/mmu.hh` and locate the declaration of `simple_mmap` in line 728. This declaration describes a simplified version of Linux's `mmap` system call. Specifically, `simple_mmap` only deals with virtual memory and it cannot map the contents of actual files.

The specification of the function is described in its comments. Your task is to implement its definition.

Testing

To test your implementation you can uncomment some or all of the tests in `test/test-all.cc`. These tests are defined in order of complexity. The test suite can be run with the following command:

```
make test
```

Submission

Once you're done, submit your modified `mmu.hh` file on isc.hevs.ch/learn. Your lab will be graded on the basis of the correctness and clarity of your implementation.