Lab #1: Basics Lab

Prof. Jae W. Lee (jaewlee@snu.ac.kr)
Department of Computer Science and Engineering
Seoul National University

TA (snu-arc-sysprog-ta@googlegroups.com)

Contents

- Goal of Lab
- Environment Overview
- Environment Setup
- Test your environment
- Grading Policy & Submission

Goal of this Lab

- **Setup your lab environment**
  - Labs 2-5 will use this environment

Environment Overview

- **You will be using a Virtual Machine Image**
  - Ubuntu 20.04 Server
  - With basic build functionality installed

- **Your assignments will be graded on this environment**
  - You may use other environments at your own risk.
  - But please check your codes on this environment before submitting.

Environment Setup

- **Please refer to the <Environment Setup Manual> on eTL for detailed information.**

- **Windows / Mac with Intel Processor:**
  - Use VirtualBox(https://www.virtualbox.org/wiki/Downloads)
  - VM Image(1.9G: https://drive.google.com/file/d/19iYCVdfY4kw2aH6n9GmwStcMpofRd22h/view?usp=drive_link)
- **Mac with Apple Silicon:**
  - Use UTM(https://mac.getutm.app/)
  - VM Image(2.7G: https://drive.google.com/file/d/1YKOiLy381JXDEwkpu8BHFbouM5t_JJcs/view?usp=drive_link)

Test Your Environment

Try the codes from lecture

- **Codes are at Github (https://github.com/SNU-ARC/2024_spring_sysprog_Lab1)**

- `$ git clone https://github.com/SNU-ARC/2024_spring_sysprog_Lab1.git`
- `$ cd 2024_spring_sysprog_Lab1`
- `$ make`

Try the codes from lecture

- **Abstraction 1: Files**
  - `$ ./abstr1_hexdump <file> [n]`
- **It will read the file and print out n characters as hex**
- **Try opening various files such as executable files**
  - Check if every file is just a chunk of bytes

Try the codes from lecture

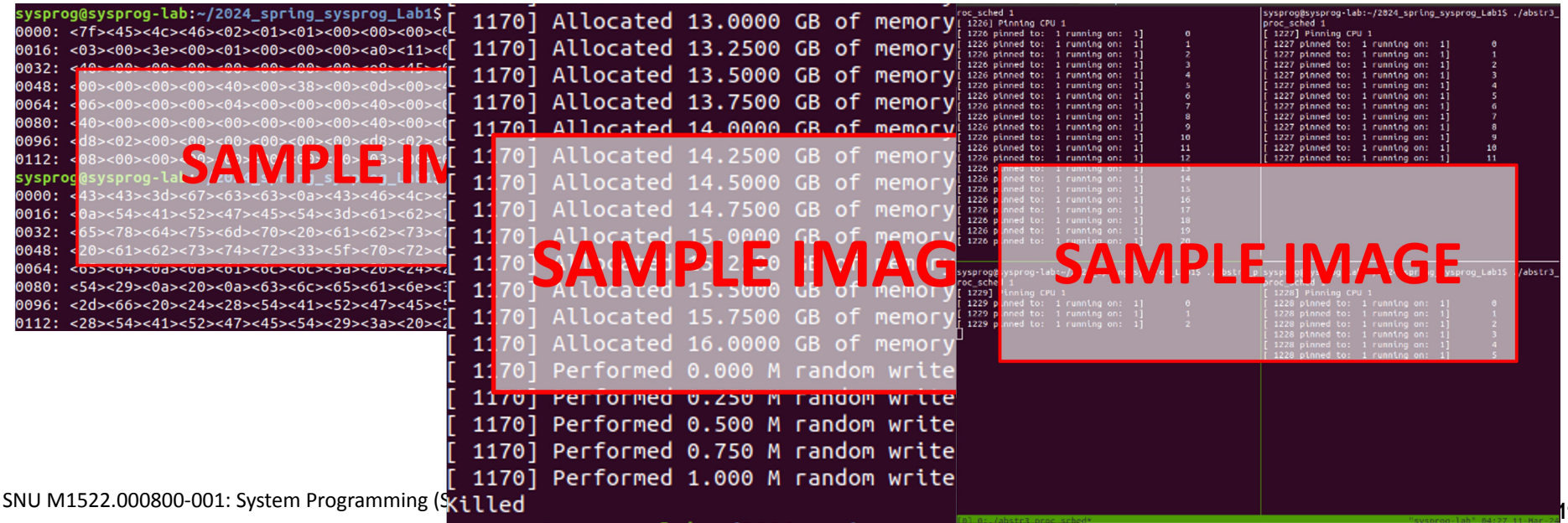
- **Abstraction 2: Virtual Memory**
  - `$ ./abstr2_mm`
- **It will allocate 16GB of memory, then do random writes**
  - VM only has 2GB of memory by default
- **Open up two terminals**
  - One with `./abstr2_mm`
  - The other with `htop`
  - Watch the memory usage change throughout the program
  - Watch when the program is killed by the OS

Try the codes from lecture

- **Abstraction 3: Process**
  - `$ ./abstr3_proc_sched <target_cpu_id>`
- **It will pin the process to one specific core**
  - The process will perform 100M loops, then print a line
- **Open up multiple terminals**
  - All `./abstr3_proc_sched` with same `<target_cpu_id>`
  - Watch the printing speed change as the number of processes change

Grading Policy & Submission

- **Take screenshots of the three programs from the lecture**
  - Compress the photos to one zip file
  - Filename should be [student id].zip (example: 2024-12345.zip)



Grading Policy & Submission

- **Submission deadline: 2024. 3. 18 (Mon) 23:59**
  - Submit via eTL

- **For late submission,**
  - 20% deduction every 24 hours
  - After next 120 hours: Submission not accepted