1. Environment Setup (2023 OS Project)

During this course, we will engage in multiple course projects to acquire practical knowledge and experience with the diverse OS features. The reference environment is Ubuntu 22.04.2 LTS.

A. Install Ubuntu on VM

If you are not running Ubuntu 22.04 natively on your computer, you may set up a VM and install Ubuntu on the VM. The following link points to instructions on how to install Ubuntu on a Hyper-V VM on Windows for your reference.

Install Ubuntu as Hyper-V Generation 2 Virtual Machine - .matrixpost.net (https://blog.matrixpost.net/install-ubuntu-as-hyper-v-generation-2-virtual-machine/)

The screen resolution can be changed via the following Windows Powershell command.

```
set-vmvideo Ubuntu -horizontalresolution:1920 -verticalresolution:1080 -resolutiontype single
```

It is fine to choose other VM platforms.

B. Install the development tools

After you install Ubuntu 22.04, you need to install the following software packages for the project.

```
sudo apt-get install git
sudo apt-get install g++
```

C. Build and trace the Hello World program

Open your editor and create the following C++ program

```
#include <stdio.h>

int main()
{
    printf("Hello World\n");
    return 0;
}
```

(main.cpp)

Compile the program and start gdb

```
g++ -g -o main main.cpp
gdb
```

In the GDB Window, type the following command and check the output as demonstrated by the screenshot blow

1. Load the main executable=> '**file main**'
2. Show the source code symbols embedded in the executable with '**list**'
3. Set a breakpoint on the print call at Line 5 with '**b 5**'
4. Start running the program with '**r**'
5. Set breakpoints on the future system call invocations with '**catch syscall**'
6. Continue program execution with '**c**'. GDB reports that it catches a system call in __GI___fstate64
7. Show the call stack with '**bt**'. It shows how the printf call (Line 5 in main.cpp) leads to __GI___fstate64
8. Use x/20i to disassemble 20 instructions from the begining of __GI___fstate64.
9. We see that the fourth instrcution of __GI___fstate64 is the **syscall** instruction.

<Exercise and Submission Checklist>

Now, please find another library call (other than **printf**) and shows the call stack leading to the '**syscall**' instruction.

Prepare a report in PDF format with the following items

A1. Description of your setup and your experiment steps.

A2. Results of your experiment. You can use screenshots, copies ot the output messages from the tools, or recording a video (need to supply the URL link to your video and make sure it is accessible by the TA)

A3. Your source code (a zipped file)

Reference

https://blog.xuite.net/yh96301/blog/63289807-VMware+Player+7.0%E4%B8%8B%E8%BC%89%E8%88%87%E5%AE%89%E8%A3%9D

Download and Install VMware Workstation Player/Pro (16/15/14) (minitool.com)

Installing Linux Ubuntu Version 20.04 in VMware Workstation Player - Windows 11 Installation Guides (dellwindowsreinstallationguide.com)

VMware Workstation 17 Player安装Ubuntu 22.04

Intel® 64 and IA-32 Architectures Software Developer Manuals