

- 操作系统实验2022
- 2022-9-19

1. 实验环境

1.1. 实验环境设备

- 自制OS的CPU: `Intel 80386`
- 模拟80386平台的虚拟机: `QEMU` (类似 NEMU, 提供模拟硬件平台)
- 交叉编译的编译器: `GCC`
- 调试工具: `GDB`
- `QEMU`, `GCC`, `GDB`的运行平台: `Ubuntu18.04`
- 编程语言: `C`, `x86 Assembly`

1.2. 实验环境搭建

方案一：使用虚拟机（推荐）

我们为同学们准备好了实验环境: `Ubuntu18.04` + `配套工具链`

我们将该环境打包成了 `OVF` 文件, 同学们只需要将该文件导入虚拟机软件, 就可直接进入实验。

如何导入

步骤1. 下载虚拟机软件并安装（三选一即可）

windows环境下:

软件名称	评价
<code>vmware workstation player</code>	免费软件, 支持文件拖拽 (推荐)

软件名称	评价
vmware workstation pro	付费软件，相比vmware player提供更多专业功能，如有办法安装可以替代vmware player
virtual box	开源免费软件，不支持文件拖拽，性能较差

mac环境下：

virtual box 或 vmware fusion

步骤2.从NJU box 下载压缩包，并解压

解压后的文件夹中有 `oslab.ovf` 和 `oslab-disk1.vmdk` 两个文件，其中 `oslab.ovf` 是虚拟机的导入配置文件，`oslab-disk1.vmdk` 是虚拟机的虚拟磁盘，虚拟机的文件都存储在这个虚拟磁盘中。

步骤3.将oslab.ovf导入虚拟机

- `vmware` 选择 打开虚拟机；`virtualbox` 在 管理 中选择 导入虚拟电脑
- 进入解压缩的目录，选择 `oslab.ovf`
- 按照提示配置导入信息（可能需要**选定非商业用途**）
- 导入

步骤4.选择导入的虚拟机，点击运行按钮（绿三角）运行虚拟机。

用户名：`oslab`

密码：`oslab2022`

至此配置完成，开始实验

方案二：手工配置（不推荐，中途可能会遇到各种不知名问题）

注：强烈建议使用Ubuntu18.04, 其他的发行版本不能保证实验在该版本上成功进行。

```
$sudo apt-get update
$sudo apt-get upgrade
$sudo apt-get install qemu-system-x86
$sudo apt-get install qemu-system-i386 vim gcc binutils
make git
```

如果有同学使用的是amd64架构，且在代码中使用了标准库，gcc使用-m32 编译选项时需要进行额外配置

第一步：确认64位架构的内核

```
$dpkg --print-architecture
amd64
```

第二步：确认打开了多架构支持功能

```
$dpkg --print-foreign-architectures
i386
```

有反馈如上说明已打开，否则需要手动打开

```
$sudo dpkg --add-architecture i386
$sudo apt-get update
$sudo apt-get dist-upgrade
```

这样就拥有了64位系统对32位的支持

最后安装gcc-multilib 和 g++-multilib 。

它们是用于交叉编译，即编译源代码以在不同于本地的处理器架构上运行。例如，如果你在 64 位 Ubuntu 上运行，并希望编译一个程序以在 32 位 Ubuntu（或 ARM 等）上运行，你就需要这样做

```
$sudo apt-get install gcc-multilib g++-multilib
```

gdb对Ubuntu18.04的版本支持只到8.X,

```
$gdb -v
```

而实验需要gdb 10.0+, 需要手动编译并安装gdb

```
$shell apt-get install libncurses5-dev  
$wget -c https://ftp.gnu.org/gnu/gdb/gdb-10.1.tar.xz  
$xz -d gdb-10.1.tar.xz  
$tar -xvf gdb-10.1.tar  
$cd gdb-10.1/  
$./configure --enable-tui  
$make -j32  
$make install
```

安装成功后, 执行

```
$gdb -v
```

版本应显示10.1

至此配置完成, 开始实验