

- 操作系统实验2022
- 2022-9

# 1. 实验环境

---

## 1.1. 实验环境设备

---

- 自制OS的CPU: Intel 80386 (32bits)
- 模拟80386平台的虚拟机: QEMU
- 交叉编译的编译器: GCC
- 调试工具: GDB
- QEMU, GCC, GDB的运行平台: Linux (建议Ubuntu18.04)
- 编程语言: C, x86 Assembly

## 1.2. 实验环境搭建

---

我们已经将所有的实验要求的虚拟机环境封装起来, 下载链接

<https://box.nju.edu.cn/f/6d07a3375b9042b8b2a5/?dl=1>

下载好虚拟机管理软件后, 直接导入其中的文件即可 (内附有更详细的说明), 再也不用麻烦的手工配置啦!!

### 前置准备

在物理机或 VMware、VirtualBox 虚拟机软件上安装发行版 Linux 系统, 例如 Debian, Ubuntu, Arch, Fedora (x86, amd64 架构皆可), 在安装完成的 Linux 上安装 QEMU, Vim, GCC, GDB, Binutils, Make, Git 等工具, 以 Ubuntu 为例

```
$sudo apt-get update
$sudo apt-get upgrade
$sudo apt-get install qemu-system-x86 vim gcc gdb 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
```

如果输出反馈则说明已打开，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 。

它们是由于交叉编译，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
```

## 版本管理

我们推荐使用Git进行项目版本管理，在整个OS实验过程中，你可能会尝试多个想法实现课程需求，也可能因为一条路走不通选择另外的思路，这时候Git就为你提供了一个在不同版本之间穿梭的机器，也可能成为你调试不通时的**后悔药**

安装好git之后，你需要先进行配置工作

```
$git config --global user.name "Zhang San" # your name
$git config --global user.email "zhangsan@foo.com" # your email
```

现在你可以使用git了，你需要切换到项目目录，然后输入

```
$git init
...
$git add file.c
...
$git commit -m "this is a comment, you can modify me"
...
```

这里只是简单的给出一些示例，具体的 git 使用方法需要同学们自行学习

## 1.3. 代码运行与调试

利用QEMU模拟80386平台，运行自制的操作系统镜像os.img

```
$qemu-system-i386 os.img
```

利用QEMU模拟80386平台，Debug自制的操作系统镜像os.img，选项-s在TCP的1234端口运行一个gdbserver，选项-S使得QEMU启动时不运行80386的CPU

```
$qemu-system-i386 -s -S os.img
```

**另开一个shell**，启动 GDB，连接上述 gdbserver，在程序计数器 0x7c00 处添加断点，运行80386的CPU，显示寄存器信息，单步执行下一条指令

备注：之所以必须是 0x7c00，是因为 BIOS就是将 MBR 读入0x7C00 地址，然后从此处开始进行后续的引导的。这个地址属于BIOS规范中定义的内容

```
$gdb
$(gdb) target remote localhost:1234
...
$(gdb) b *0x7c00
...
$(gdb) continue
...
$(gdb) info registers
...
$(gdb) si
...
```

实际上在后续的实验中，你可能很难通过程序计数器添加断点，而我们生成的可执行文件也可能没有符号表信息，这时候就需要使用

```
$(gdb) file example # 可执行文件
```

这样就支持使用行号、函数名等方式添加断点，具体内容请参考gdb手册或自行查阅资料

## 2. 相关资料

---

### 2.1. 相关网站

---

- 维基网站 <http://wiki.osdev.org>
- 问答网站 <http://stackoverflow.com>

### 2.2. 相关手册

---

- Intel 80386 Programmer's Reference Manual
- GCC 4.4.7 Manual
- GDB User Manual
- GNU Make Manual
- System V ABI
- [Linux Manual Page](#)

## 3. 实验列表

---

### 3.1. Lab1系统引导

---

实现一个简单的引导程序

### 3.2. Lab2系统调用

---

实现一个简单的系统调用

### 3.3. Lab3进程切换

---

实现一个简单的任务调度

### 3.4. Lab4进程同步

---

实现一个简单的同步机制

### 3.5. Lab5文件系统

---

实现一个简单的文件系统

## 4. 作业规范与提交

---

### 4.1. 作业规范

---

- **学术诚信:** 如果你确实无法完成实验, 你可以选择不提交, **作为学术诚信的奖励, 你将会获得10%的分数;** 但若发现抄袭现象, 抄袭双方(或团体)在本次实验中得0分.
- 实验源码提交前需清除编译生成的临时文件, 虚拟机镜像等无关文件

- 请你在实验截止前务必确认你提交的内容符合要求(格式, 相关内容等), 你可以下载你提交的内容进行确认. 如果由于你的原因给我们造成了不必要的麻烦, 视情况而定, 在本次实验中你将会被扣除该次实验得分的部分分数, 最高可达50%
- 实验不接受迟交, 一旦迟交按**学术诚信**给分
- **本实验给分最终解释权归助教所有**

## 4.2. 提交格式

每次实验的框架代码结构如下

```
labX-YYYYYYYYY.zip          #修改文件夹名称
├── labX
│   ├── Makefile
│   └── ...
└── report
    └── YYYYYYYY.pdf          # 替换为自己的实验报告
```

- labX中的X代表实验序号, 如lab1, labX/目录存放最终版本的源代码、编译脚本
- **report/目录存放实验报告, 要求为pdf格式**
- 在提交作业之前先将文件名称更改为正确的格式, 打包为一个单独的zip包
- **压缩包以学号命名, 例如 lab1-201300000.zip**是符合格式要求的压缩包名称
- 为了防止出现编码问题, 压缩包中的所有文件名都**不要包含中文**
- 我们只接受pdf格式, **命名只含学号的实验报告**, 不符合格式的实验报告将视为没有提交报告.
  - 例如 **201300000.pdf** 是符合格式要求的实验报告,
  - 但 201300000.docx 和 201300000张三实验报告.pdf 不符合要求
- 作业提交网站 <http://cslabcms.nju.edu.cn>

## 4.3. 实验报告内容

你**必须**在实验报告中描述以下内容

- **姓名、学号、邮箱**等信息, 方便我们及时给你一些反馈信息
- 实验进度。简单描述即可, 例如"我完成了所有内容", "我只完成了xxx"。缺少实验进度的描述, 或者描述与实际情况不符, 将被视为没有完成本次实验
- 实验结果。贴图或说明都可, 不需要太复杂, 确保不要用其他同学的结果, 否则以抄袭处理
- 实验修改的代码位置, 简单描述为完成本次实验, 修改或添加了哪些代码。不需要贴图或一行行解释, 大致的文件和函数定位就行

你可以**自由选择**报告的其它内容。你不必详细地描述实验过程, 但我们鼓励你在报告中描述如下内容:

- 你遇到的问题和对这些问题的思考
- 对讲义或框架代码中某些思考题的看法
- 或者你的其它想法, 例如实验心得, 对提供帮助的同学的感谢等

认真描述实验心得和想法的报告将会获得分数的奖励; 思考题选做, 完成了也不会得到分数的奖励, 但它们是经过精心准备的, 可以加深你对某些知识的理解和认识。如果你实在没有想法, 你可以提交一份不包含任何想法的报告, 我们不会强求。但请**不要**

- 大量粘贴讲义内容
- 大量粘贴代码和贴图, 却没有相应的详细解释(让我们明显看出来是凑字数的)

来让你的报告看起来十分丰富, 编写和阅读这样的报告毫无任何意义, 你也不会因此获得更多的分数, 同时还可能带来扣分的可能。