

# Lab 0

## 前言

由于使用了wsl(windows subsystem for linux, 即适用于windows的linux子系统)来配置环境, 这种环境可能是大多数人没使用过的, 而且wsl1只是使用windows NT实现的linux内核, 通过pipe转发linux命令到windows来完成命令实现并方便交互。

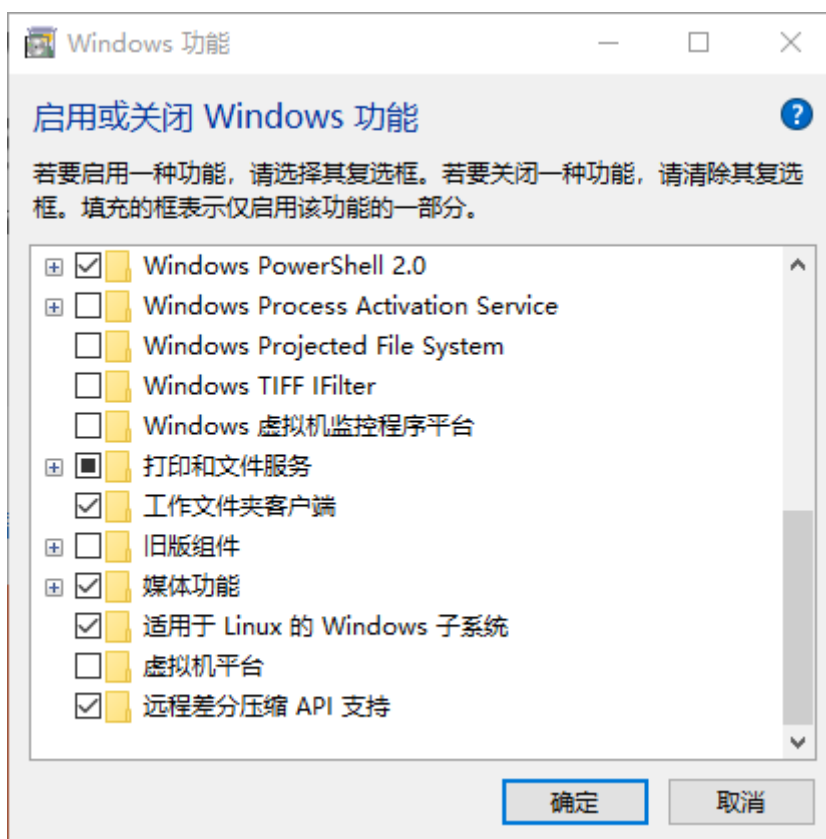
除此以外, MIT 6.828使用qemu虚拟机, 但是是专门定制的, 所以需要去MIT-pdos的github账户下载或者git同步, 而且配置会相对比较花费时间, 容易出现奇怪的bug。

综合上述两个原因, 专门花费一章来讲述wsl配置和qemu的安装, 顺便暗示一下这个Lab中的Lab。自然地, Lab1中也不再讲述这部分内容。

## Windows子系统的配置

Windows子系统是win10在1703之后的版本, 具体配置可以在[微软官网](#)找到, 这里讲述一下个人的配置过程。

### 安装wsl



这个功能也可以使用管理员下的Powershell完成。

```
1 | Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
```

但是不论使用哪种方式, 都需要重启系统已完成最终的功能配置。然后在Windows商店搜索, 下载并安装Ubuntu-18.04 LTS。



单击这个装好的应用，就可以打开终端，这个相当于安装系统的功能。安装需要一段时间，之后输入账号密码就可以使用了，由于是本地系统，所以设置简单密码也不会被攻击。如果跳过第一步，你会收到一条消息，让你回去把那个打开在来打开这个应用。

下面是安装完成的截图。

```
ivy233@DESKTOP-LVJJABH: ~
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: ivy233
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ivy233@DESKTOP-LVJJABH:~$ sudo apt-get
[sudo] password for ivy233:
apt 1.6.10 (amd64)
Usage: apt-get [options] command
       apt-get [options] install|remove pkg1 [pkg2 ...]
       apt-get [options] source pkg1 [pkg2 ...]

apt-get is a command line interface for retrieval of packages
and information about them from authenticated sources and
for installation, upgrade and removal of packages together
with their dependencies.

Most used commands:
update - Retrieve new lists of packages
upgrade - Perform an upgrade
install - Install new packages (pkg is libc6 not libc6.deb)
remove - Remove packages
purge - Remove packages and config files
autoremove - Remove automatically all unused packages
dist-upgrade - Distribution upgrade, see apt-get(8)
dselect-upgrade - Follow dselect selections
build-dep - Configure build-dependencies for source packages
clean - Erase downloaded archive files
autoclean - Erase old downloaded archive files
check - Verify that there are no broken dependencies
source - Download source archives
download - Download the binary package into the current directory
changelog - Download and display the changelog for the given package

See apt-get(8) for more information about the available commands.
Configuration options and syntax is detailed in apt.conf(5).
Information about how to configure sources can be found in sources.list(5).
Package and version choices can be expressed via apt_preferences(5).
Security details are available in apt-secure(8).
                                     This APT has Super Cow Powers.
ivy233@DESKTOP-LVJJABH:~$
```

## 配置wsl

和Ubuntu一样，内置的是ubuntu自带软件源，由于下载特别慢，所以需要换源（如果你足够牛皮，可以考虑配置apt代理）。

除此以外，由于windows另外一个开源工具Terminal特别好用，笔者选用此终端完成剩下的内容，如果不想用的话，也可以使用原来的应用。

命令如下：

```
1 | sudo vi /etc/apt/sources.list
```

然后把这个换成比较著名的国内源即可，如果不知道的可以按照[这个](#)来配置。之后和Ubuntu一样更新软件。

```
1 | sudo apt-get update
2 | sudo apt-get upgrade
```

## MIT 6.828 环境配置

如果你直接开始看这里，推荐回头看看wsl的配置。

### 安装依赖

用这个命令安装qemu需要的依赖。原教程有libSDL-dev，这个主要用于产生qemu的界面。Git工具一般wsl会自带，这里也带上。

```
1 | sudo apt-get install git build-essential binutils g++-multilib libgtk2.0-dev libtool-bin
```

### 安装qemu

这一步可以和第一步同时进行。使用这个命令来获得qemu的的最新包。这一步将会花费比较多的时间。

```
1 | git clone https://github.com/mit-pdos/6.828-qemu.git qemu
```

但是一般速度很慢，建议配置代理。

```
1 | git config --global https.proxy socks://127.0.0.1:1080
2 | git config --global http.proxy socks://127.0.0.1:1080
```

不过千万要注意的是：不能再/mnt目录下完成。如果在VMWare下，不能在/mnt/hgfs中完成。下载完成之后需要make一下。

```
1 | cd qemu
2 | ./configure --disable-kvm --disable-werror --target-list="i386-softmmu x86_64-softmmu" --prefix=/usr/local/qemu
3 | sudo make
4 | sudo make install
```

### 进入lab虚拟机

在进入系统之前，需要使用git同步lab：

```
1 | git clone https://pdos.csail.mit.edu/6.828/2018/jos.git lab
```

然后打开lab/conf/env.mk，修改最后一行的qemu目录，需要和之前的匹配。

到这一步之前都可以同步执行，甚至按照教程来的话有可能第三步首先执行完成。之后输入

```
1 | cd lab
2 | make qemu-nox
```

可以看到这个界面时，配置完成。啥都没有是正常的，因为wsl确实没有显示输出，需要安装XShell才有，因此这里选择qemu-nox，而不是教程所说的qemu。

```
ivy233@DESKTOP-LVJJABH: ~/qemu x + v - □ ×
set -e; for x in da      en-gb et fr      fr-ch is lt modifiers no pt-br sv ar      de      e
n-us fi fr-be hr      it lv nl      pl ru      th common de-ch es      fo fr-ca hu      j
a mk nl-be      pt sl      tr bepo      cz; do \
    install -c -m 0644 /home/ivy233/qemu/pc-bios/keymaps/$x "/usr/local/qemu/share/qemu/keymap
s"; \
done
install -c -m 0644 /home/ivy233/qemu/trace-events "/usr/local/qemu/share/qemu/trace-events"
for d in i386-softmmu x86_64-softmmu; do \
make --no-print-directory BUILD_DIR=/home/ivy233/qemu TARGET_DIR=$d/ -C $d install || exit 1 ; \
done
install -d -m 0755 "/usr/local/qemu/bin"
libtool --quiet --mode=install install -c -m 0755 qemu-system-i386 "/usr/local/qemu/bin"
strip "/usr/local/qemu/bin/qemu-system-i386"
install -d -m 0755 "/usr/local/qemu/bin"
libtool --quiet --mode=install install -c -m 0755 qemu-system-x86_64 "/usr/local/qemu/bin"
strip "/usr/local/qemu/bin/qemu-system-x86_64"
ivy233@DESKTOP-LVJJABH:~/qemu$ ls -ah /usr/local/
. .. bin etc games include lib man qemu sbin share src
ivy233@DESKTOP-LVJJABH:~/qemu$ /usr/local/qemu/bin/qemu-system-i386

(qemu-system-i386:1342): Gtk-WARNING **: 15:10:11.421: cannot open display:
ivy233@DESKTOP-LVJJABH:~/qemu$
```

```
ivy233@DESKTOP-LVJJABH: ~/lab x + v - □ ×
ivy233@DESKTOP-LVJJABH:~/qemu$ cd ../lab
ivy233@DESKTOP-LVJJABH:~/lab$ make qemu-nox
+ as kern/entry.S
+ cc kern/entrypgdir.c
+ cc kern/init.c
+ cc kern/console.c
+ cc kern/monitor.c
+ cc kern/printf.c
+ cc kern/kdebug.c
+ cc lib/printfmt.c
+ cc lib/readline.c
+ cc lib/string.c
+ ld obj/kern/kernel
ld: warning: section `bss' type changed to PROGBITS
+ as boot/boot.S
+ cc -Os boot/main.c
+ ld boot/boot
boot block is 390 bytes (max 510)
+ mk obj/kern/kernel.img
sed "s/localhost:1234/localhost:26000/" < .gdbinit.tmpl > .gdbinit
***
*** Use Ctrl-a x to exit qemu
***
/usr/local/qemu/bin/qemu-system-i386 -nographic -drive file=obj/kern/kernel.img,index=0,media=disk
,format=raw -serial mon:stdio -gdb tcp::26000 -D qemu.log
6828 decimal is XXX octal!
entering test_backtrace 5
entering test_backtrace 4
entering test_backtrace 3
entering test_backtrace 2
entering test_backtrace 1
entering test_backtrace 0
leaving test_backtrace 0
leaving test_backtrace 1
leaving test_backtrace 2
leaving test_backtrace 3
leaving test_backtrace 4
leaving test_backtrace 5
Welcome to the JOS kernel monitor!
Type 'help' for a list of commands.
K>
```

## 可能遇到的问题

关于大部分遇到的问题，网上都有解答，甚至有网页专门写了网页来解答。但是我在配置的时候遇到了一个网上搜不到的问题。

```
ivy233@DESKTOP-9G8VB9R:~/qemu$ ./configure --disable-kvm --disable-werror --target-list="i386-soft
prefix=/usr/local/qemu
: not found: 5: ./configure:
: bad variable nameet: GREP_OPTIONS
```

如果遇到了这个问题，请重装windows 10。我试过给环境变量加上GREP\_OPTIONS，注释那一行，重置wsl，重置wsl功能，都是无效的。

在写本文的时候出现了一个新的问题：内容没有截下来，具体表现为没有这个命令，然后看一下文件夹分布

```
1 /mnt/c/Users/95225$ cd ~
2 ~$ ls -ah
3 . .bash_history .bashrc .profile .viminfo qemu
4 .. .bash_logout .gitconfig .sudo_as_admin_successful lab
5 ~$ ls -ah /usr
6 . .. bin games include lib lib32 libx32 local sbin share src
7 ~$ ls -ah /usr/local/
8 . .. bin etc games include lib man sbin share src
9 ~$ ls -ah /usr/local/bin
10 . ..
```

解决方法就是把第三步重新做一遍。

## 小结

Windows下的wsl是相当好用的工具，在windows上提供了和linux命令行一样的体验，同时保持了小体积和系统级多开(微软商店提供了Ubuntu和Ubuntu18.04, Ubuntu16.04三个版本)，提供了很多的便利。

与此同时，qemu的配置比较复杂，完成这一部分本身就可以写一份比较复杂的实验报告了，本身也是很有成就感的。

不过不要忘了，这里只是第一步，虽然千里之行始于足下，但是这条路并不如想象中平坦。