OS-lab0

安装Docker

根据docker的安装教程安装新版本的Docker

```
br_win@SPARKLE-AIR:~/linux_kernel/linux-6.0-rc5$ sudo apt install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-ce-cli docker-ce-rootless-extras docker-scan-plugin pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras docker-scan-plugin pigz slirp4netns
0 upgraded, 7 newly installed, 0 to remove and 225 not upgraded.
Need to get 102 MB of archives.
After this operation, 397 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 https://mirrors.aliyun.com/ubuntu focal/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 https://download.docker.com/linux/ubuntu focal/stable amd64 containerd.io amd64 1.6.8-1 [28.1 MB]
Get:3 https://mirrors.aliyun.com/ubuntu focal/universe amd64 slirp4netns amd64 0.4.3-1 [74.3 kB]
6% [2 containerd.io 327 kB/28.1 MB 1%]
```

安装完成

```
To get more help with docker, check out our guides at https://u
br_win@SPARKLE-AIR:~/linux_kernel/linux-6.0-rc5$ docker -v
Docker version 20.10.18, build b40c2f6
```

搭建Docker环境

按照实验步骤进行，成功搭建了Docker环境。

期间出现的Error是由于在之前的命令中已经有了oslab容器的存在，因此无法新建，当使用rm命令删除oslab容器之后，即可正常执行最后一条命令。

```
br_win@SPARKLE-AIR: ~
oslab         2021      8c15472cd111   7 months ago    3.63GB
hello-world   latest    feb5d9fea6a5   11 months ago   13.3kB
br_win@SPARKLE-AIR:~$ docker run --name oslab -it oslab:2021 bash
root@3f36fce19ef5:/# exit
br_win@SPARKLE-AIR:~$ docker start oslab
oslab
br_win@SPARKLE-AIR:~$ ps
  PID TTY          TIME CMD
   10 pts/0    00:00:00 bash
 5917 pts/0    00:00:00 ps
br_win@SPARKLE-AIR:~$ docker ps
CONTAINER ID   IMAGE        COMMAND   CREATED         STATUS         PORTS     NAMES
3f36fce19ef5   oslab:2021   "bash"    20 seconds ago   Up 7 seconds             oslab
br_win@SPARKLE-AIR:~$ docker exec -it oslab bash
root@3f36fce19ef5:/# exit
br_win@SPARKLE-AIR:~$ docker run --name oslab -it -v ${HOME}:/have-fun-debugging oslab:2021 bash
docker: Error response from daemon: Conflict. The container name "/oslab" is already in use by container "3f36fce19ef5264846481715bce763f710cc0479765bc9a2b85c8a9e615ec99a".
 You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
br_win@SPARKLE-AIR:~$ docker -v ${HOME}:/have-fun-debugging oslab:2021 bash
Docker version 20.10.18, build b40c2f6
br_win@SPARKLE-AIR:~$ docker kill oslab
oslab
br_win@SPARKLE-AIR:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
br_win@SPARKLE-AIR:~$ docker run --name oslab -it -v ${HOME}:/have-fun-debugging oslab:2021 bash
docker: Error response from daemon: Conflict. The container name "/oslab" is already in use by container "3f36fce19ef5264846481715bce763f710cc0479765bc9a2b85c8a9e615ec99a".
 You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
br_win@SPARKLE-AIR:~$ docker rm oslab
oslab
br_win@SPARKLE-AIR:~$ docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED         STATUS                     PORTS     NAMES
ce0e51cd5950   hello-world   "/hello"   8 minutes ago   Exited (0) 8 minutes ago             eager_pasteur
br_win@SPARKLE-AIR:~$ docker rm hello-world
Error: No such container: hello-world
br_win@SPARKLE-AIR:~$ docker run --name oslab -it -v ${HOME}:/have-fun-debugging oslab:2021 bash
root@6eaf23b100d7:/# exit
br_win@SPARKLE-AIR:~$
```

获得Linux源码

在Linux官网下载并解压Linux源码

```
br_win@SPARKLE-AIR:~/linux_kernel$ ls
linux-6.0-rc5  linux-6.0-rc5.tar.gz  linux-6.0-rc5.tar.gz:Zone.Identifier  rootfs.img
br_win@SPARKLE-AIR:~/linux_kernel$ cd linux-6.0-rc5/
br_win@SPARKLE-AIR:~/linux_kernel/linux-6.0-rc5$ ls
COPYING  Documentation  Kconfig   MAINTAINERS  README  block  crypto   fs       init      ipc     lib  net      scripts   sound  usr
CREDITS  Kbuild         LICENSES  Makefile     arch    certs  drivers  include  io_uring  kernel  mm   samples  security  tools  virt
br_win@SPARKLE-AIR:~/linux_kernel/linux-6.0-rc5$
```

编译Linux内核

按照Manual生成配置并对Linux内核进行编译

```
root@6eaf23b100d7:/have-fun-debugging# pwd
/have-fun-debugging
root@6eaf23b100d7:/have-fun-debugging# ls
C  ImageAnnotation  linux-6.0-rc5.tar.gz:Zone.Identifier  linux_kernel  oslab.tar  oslab.tar:Zone.Identifier  server_xlab
root@6eaf23b100d7:/have-fun-debugging# cd linux_kernel/
root@6eaf23b100d7:/have-fun-debugging/linux_kernel# ls
linux-6.0-rc5  linux-6.0-rc5.tar.gz  linux-6.0-rc5.tar.gz:Zone.Identifier  rootfs.img
root@6eaf23b100d7:/have-fun-debugging/linux_kernel# cd linux-6.0-rc5
root@6eaf23b100d7:/have-fun-debugging/linux_kernel/linux-6.0-rc5# ls
COPYING  Documentation  Kconfig   MAINTAINERS  README  block  crypto   fs       init      ipc     lib  net      scripts   sound  usr
CREDITS  Kbuild         LICENSES  Makefile     arch    certs  drivers  include  io_uring  kernel  mm   samples  security  tools  virt
root@6eaf23b100d7:/have-fun-debugging/linux_kernel/linux-6.0-rc5# make ARCH=riscv CROSS_COMPILE=riscv64-unknown-linux-gnu- defconfig
  HOSTCC  scripts/basic/fixdep
  HOSTCC  scripts/kconfig/conf.o
  HOSTCC  scripts/kconfig/confdata.o
  HOSTCC  scripts/kconfig/expr.o
  LEX     scripts/kconfig/lexer.lex.c
  YACC    scripts/kconfig/parser.tab.[ch]
  HOSTCC  scripts/kconfig/lexer.lex.o
  HOSTCC  scripts/kconfig/menu.o
  HOSTCC  scripts/kconfig/parser.tab.o
  HOSTCC  scripts/kconfig/preprocess.o
  HOSTCC  scripts/kconfig/symbol.o
  HOSTCC  scripts/kconfig/util.o
  HOSTLD  scripts/kconfig/conf
*** Default configuration is based on 'defconfig'
#
# configuration written to .config
#
root@6eaf23b100d7:/have-fun-debugging/linux_kernel/linux-6.0-rc5# make ARCH=riscv CROSS_COMPILE=riscv64-unknown-linux-gnu- -j$(nproc)
```

编译完成

```
root@6eaf23b100d7: /have-fu
  LD [M]  net/ipv4/netfilter/ip_tables.ko
  LD [M]  net/ipv4/netfilter/ipt_REJECT.ko
  LD [M]  net/ipv4/netfilter/iptable_filter.ko
  LD [M]  net/ipv4/netfilter/iptable_mangle.ko
  LD [M]  net/ipv4/netfilter/iptable_nat.ko
  LD [M]  net/ipv4/netfilter/nf_defrag_ipv4.ko
  LD [M]  net/ipv4/netfilter/nf_reject_ipv4.ko
  LD [M]  net/ipv4/udp_tunnel.ko
  LD [M]  net/ipv6/ip6_udp_tunnel.ko
  LD [M]  net/ipv6/netfilter/ip6_tables.ko
  LD [M]  net/ipv6/netfilter/ip6t_ipv6header.ko
  LD [M]  net/ipv6/netfilter/ip6t_REJECT.ko
  LD [M]  net/ipv6/netfilter/ip6table_filter.ko
  LD [M]  net/ipv6/netfilter/ip6table_mangle.ko
  LD [M]  net/ipv6/netfilter/nf_reject_ipv6.ko
  LD [M]  net/llc/llc.ko
  LD [M]  net/ipv6/netfilter/nf_defrag_ipv6.ko
  LD [M]  net/netfilter/ipvs/ip_vs.ko
  LD [M]  net/netfilter/ipvs/ip_vs_rr.ko
  LD [M]  net/netfilter/nf_conntrack.ko
  LD [M]  net/netfilter/nf_conntrack_ftp.ko
  LD [M]  net/netfilter/nf_conntrack_tftp.ko
  LD [M]  net/netfilter/nf_log_syslog.ko
  LD [M]  net/netfilter/nf_nat.ko
  LD [M]  net/netfilter/nf_nat_ftp.ko
  LD [M]  net/netfilter/nf_nat_tftp.ko
  LD [M]  net/netfilter/x_tables.ko
  LD [M]  net/netfilter/xt_REDIRECT.ko
  LD [M]  net/netfilter/xt_MASQUERADE.ko
  LD [M]  net/netfilter/xt_addrtype.ko
  LD [M]  net/netfilter/xt_conntrack.ko
  LD [M]  net/netfilter/xt_ipvs.ko
  LD [M]  net/netfilter/xt_mark.ko
  LD [M]  net/netfilter/xt_tcpudp.ko
  LD [M]  net/netfilter/xt_nat.ko
  LD [M]  net/sched/cls_cgroup.ko
  LD [M]  net/xfrm/xfrm_algo.ko
  LD [M]  net/xfrm/xfrm_user.ko
root@6eaf23b100d7:/have-fun-debugging/linux_kernel/linux-6.0-rc5#
```

根据自身情况调整参数运行：

```
qemu-system-riscv64 -nographic -machine virt -kernel /have-fun-debugging/linux_kernel/linux-6.0-rc5/arch/riscv/boot/Image  -device virtio-blk-device,drive=hd0 -append "root=/dev/vda ro console=ttyS0"   -bios default -drive file=rootfs.img,format=raw,id=hd0
```

```
root@6eaf23b100d7:/have-fun-debugging/linux_kernel/linux-6.0-rc5# pwd
/have-fun-debugging/linux_kernel/linux-6.0-rc5
root@6eaf23b100d7:/have-fun-debugging/linux_kernel/linux-6.0-rc5# ls
COPYING        Kbuild    MAINTAINERS     README      block   drivers  init      kernel  modules-only.symvers     modules.order  scripts   tools  vmlinux
CREDITS        Kconfig   Makefile        System.map  certs   fs       io_uring  lib     modules.builtin           net            security  usr    vmlinux.o
Documentation  LICENSES  Module.symvers  arch        crypto  include  ipc       mm      modules.builtin.modinfo   samples        sound     virt   vmlinux.symvers
root@6eaf23b100d7:/have-fun-debugging/linux_kernel/linux-6.0-rc5# cd ..
root@6eaf23b100d7:/have-fun-debugging/linux_kernel# ls
linux-6.0-rc5  linux-6.0-rc5.tar.gz  linux-6.0-rc5.tar.gz:Zone.Identifier  rootfs.img
root@6eaf23b100d7:/have-fun-debugging/linux_kernel# qemu-system-riscv64 -nographic -machine virt -kernel /have-fun-debugging/linux_kernel/linux-6.0-rc5/arch/riscv/boot/Imag
e  -device virtio-blk-device,drive=hd0 -append "root=/dev/vda ro console=ttyS0"   -bios default -drive file=rootfs.img,format=raw,id=hd0
```

成功启动QEMU运行内核

```
root@6eaf23b100d7: /have-fu
[    0.424068] loop: module loaded
[    0.425027] virtio_blk virtio0: 1/0/0 default/read/poll queues
[    0.428645] virtio_blk virtio0: [vda] 32768 512-byte logical blocks (16.8 MB/16.0 MiB)
[    0.455260] e1000e: Intel(R) PRO/1000 Network Driver
[    0.455494] e1000e: Copyright(c) 1999 - 2015 Intel Corporation.
[    0.457612] ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
[    0.458519] ehci-pci: EHCI PCI platform driver
[    0.459257] ehci-platform: EHCI generic platform driver
[    0.460325] ohci_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver
[    0.461225] ohci-pci: OHCI PCI platform driver
[    0.462139] ohci-platform: OHCI generic platform driver
[    0.464301] usbcore: registered new interface driver uas
[    0.465641] usbcore: registered new interface driver usb-storage
[    0.468073] mousedev: PS/2 mouse device common for all mice
[    0.472028] goldfish_rtc 101000.rtc: registered as rtc0
[    0.472913] goldfish_rtc 101000.rtc: setting system clock to 2022-09-14T09:51:15 UTC (1663149075)
[    0.476937] cpuidle-riscv-sbi: HSM suspend not available
[    0.477915] sdhci: Secure Digital Host Controller Interface driver
[    0.478350] sdhci: Copyright(c) Pierre Ossman
[    0.478827] sdhci-pltfm: SDHCI platform and OF driver helper
[    0.480435] usbcore: registered new interface driver usbhid
[    0.481031] usbhid: USB HID core driver
[    0.484228] NET: Registered PF_INET6 protocol family
[    0.492343] Segment Routing with IPv6
[    0.492776] In-situ OAM (IOAM) with IPv6
[    0.493524] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
[    0.496800] NET: Registered PF_PACKET protocol family
[    0.498647] 9pnet: Installing 9P2000 support
[    0.500200] Key type dns_resolver registered
[    0.503359] debug_vm_pgtable: [debug_vm_pgtable         ]: Validating architecture page table helpers
[    0.517697] Legacy PMU implementation is available
[    0.557593] EXT4-fs (vda): mounted filesystem with ordered data mode. Quota mode: disabled.
[    0.558264] VFS: Mounted root (ext4 filesystem) readonly on device 254:0.
[    0.562258] devtmpfs: mounted
[    0.591457] Freeing unused kernel image (initmem) memory: 2176K
[    0.594015] Run /sbin/init as init process

Please press Enter to activate this console.
/ #
```

心得体会

这次lab0踩了很多坑，不过基本上都是踩在了一开始自己配环境的时候。前期的配环境除了我的apt源过旧的问题，更新了一下源外，并没有出其他什么大的问题，但是（怀疑）由于qemu在Ubuntu的apt源中版本过低的问题，导致在使用qeme启动时报了

> qemu-system-riscv64: Unable to load the RISC-V firmware "opensbi-riscv64-virt-fw_jump.bin"

的错误。为了防止在之后的Lab中出现一些类似于版本不兼容等难以解决的奇奇怪怪的问题，以及为了和队友的开发环境保持一致，因此还是使用了docker。

在使用gdb进行调试的时候，一直会出现start_kernel的函数无法打上断点的情况。后来是通过更换qumu的版本解决的问题。再后来得知了，只需要再gdb调试的时候加上`set riscv use-compressed-breakpoints on`，就也可以解决这个问题。