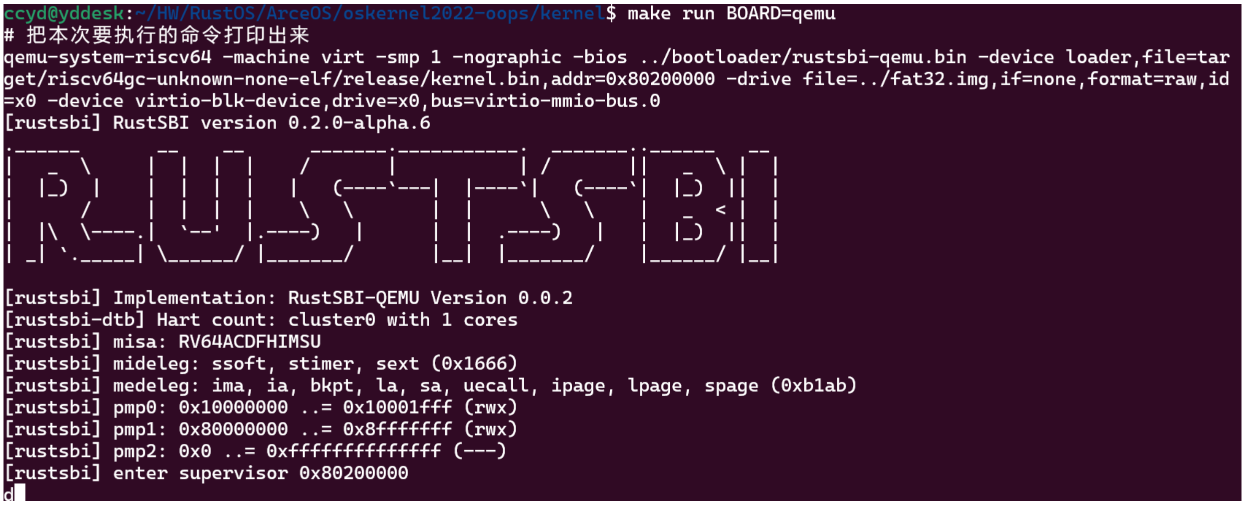
Week8进展记录

1. 哈尔滨工业大学(深圳)-Oops

构建的过程非常坎坷



```
ccyd@yddesk:~/HW/RustOS/ArceOS/oskernel2022-oops/kernel$ make run BOARD=qemu
# 把本次要执行的命令打印出来
qemu-system-riscv64 -machine virt -smp 1 -nographic -bios ../bootloader/rustsbi-qemu.bin -device loader,file=tar
get/riscv64gc-unknown-none-elf/release/kernel.bin,addr=0x80200000 -drive file=../fat32.img,if=none,format=raw,id
=x0 -device virtio-blk-device,drive=x0,bus=virtio-mmio-bus.0
[rustsbi] RustSBI version 0.2.0-alpha.6
.______       __    __      _______.___________.  _______..______   __
|   _  \     |  |  |  |    /       |           | /       ||   _  \ |  |
|  |_)  |    |  |  |  |   |   (----`---|  |----`|   (----`|  |_)  ||  |
|      /     |  |  |  |    \   \       |  |      \   \    |   _  < |  |
|  |\  \----.|  `--'  |.----)   |      |  |  .----)   |   |  |_)  ||  |
| _| `._____| \______/ |_______/       |__|  |_______/    |______/ |__|

[rustsbi] Implementation: RustSBI-QEMU Version 0.0.2
[rustsbi-dtb] Hart count: cluster0 with 1 cores
[rustsbi] misa: RV64ACDFHIMSU
[rustsbi] mideleg: ssoft, stimer, sext (0x1666)
[rustsbi] medeleg: ima, ia, bkpt, la, sa, uecall, ipage, lpage, spage (0xb1ab)
[rustsbi] pmp0: 0x10000000 ..= 0x10001fff (rwx)
[rustsbi] pmp1: 0x80000000 ..= 0x8fffffff (rwx)
[rustsbi] pmp2: 0x0 ..= 0xffffffffffffff (---)
[rustsbi] enter supervisor 0x80200000
d
```

2. 初赛testsuit编译

1. 编译riscv-toolchain

因为要用到xxx-linux-elf-xx，Tutorial里预编译的版本不能使用。

2. 编译riscv-linux-rootfs

没有成功，卡在

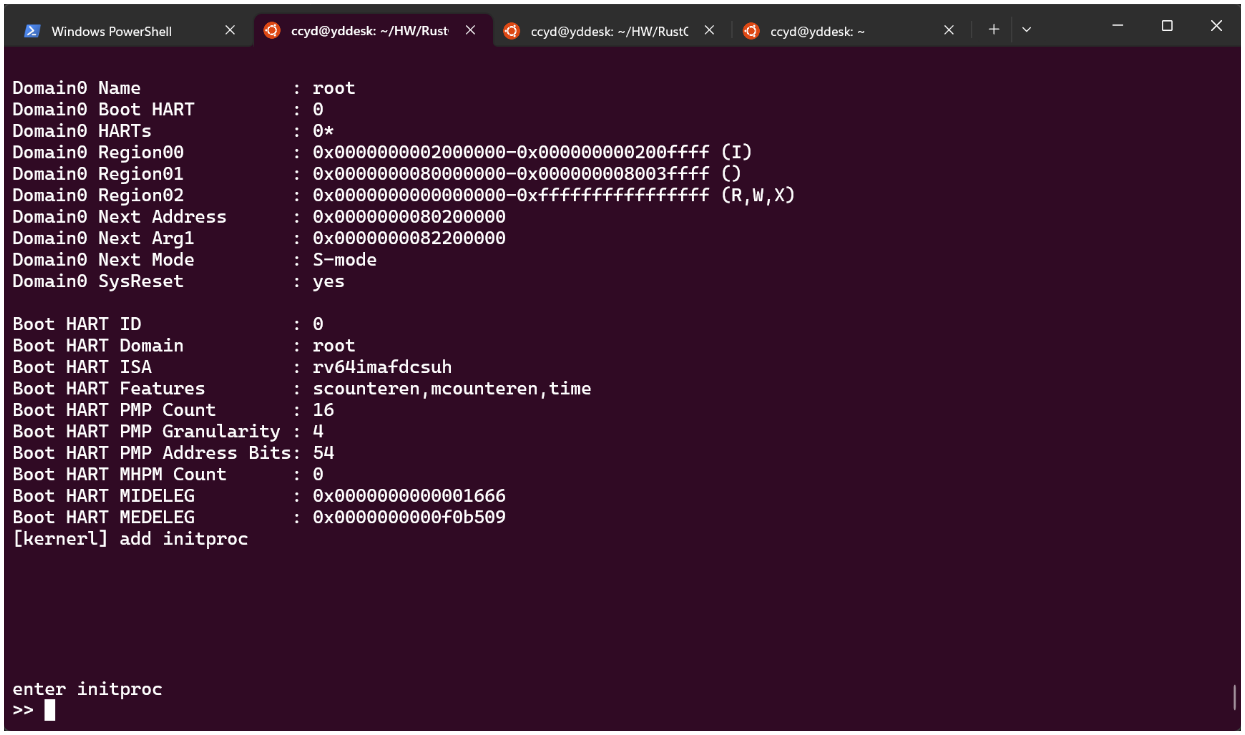
```
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

mkfs.fat 4.1 (2017-01-24)
*** failed to create riscv64-rootfs.bin
```

3.编译riscv-syscalls-testing

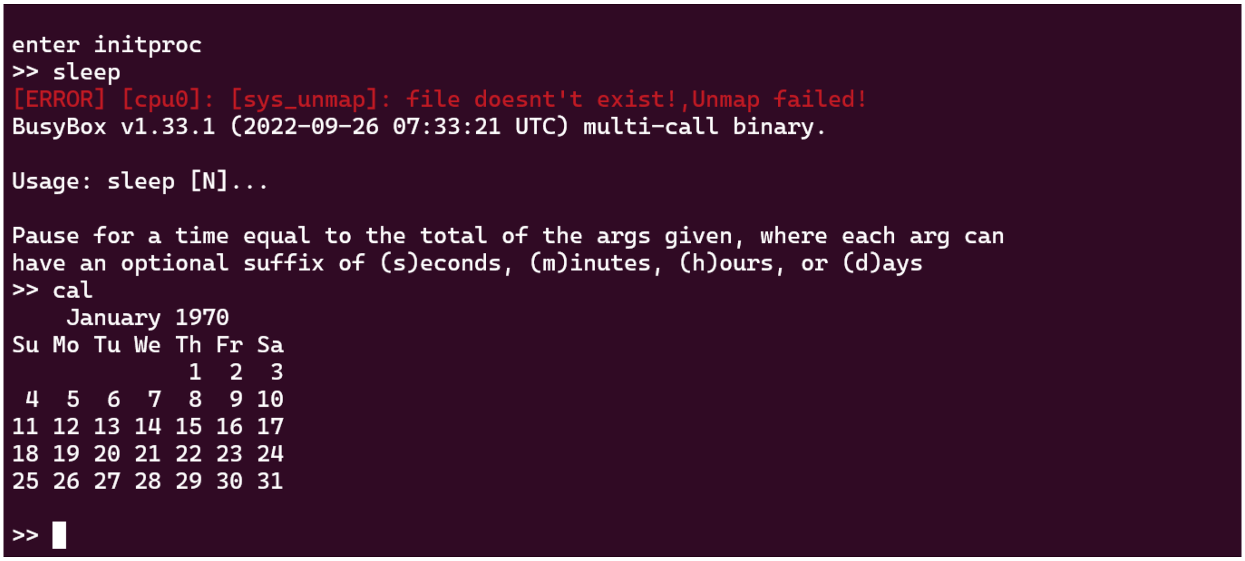
按README做就好。

3. fat32.img制作

制作好之后加载到Oops运行



```
Domain0 Name              : root
Domain0 Boot HART         : 0
Domain0 HARTs             : 0*
Domain0 Region00          : 0x0000000002000000-0x000000000200ffff (I)
Domain0 Region01          : 0x0000000080000000-0x000000008003ffff ()
Domain0 Region02          : 0x0000000000000000-0xffffffffffffffff (R,W,X)
Domain0 Next Address      : 0x0000000080200000
Domain0 Next Arg1         : 0x0000000082200000
Domain0 Next Mode         : S-mode
Domain0 SysReset          : yes

Boot HART ID              : 0
Boot HART Domain          : root
Boot HART ISA             : rv64imafdcsuh
Boot HART Features        : scounteren,mcounteren,time
Boot HART PMP Count       : 16
Boot HART PMP Granularity : 4
Boot HART PMP Address Bits: 54
Boot HART MHPM Count      : 0
Boot HART MIDELEG         : 0x0000000000001666
Boot HART MEDELEG         : 0x0000000000f0b509
[kernerl] add initproc






enter initproc
>> 
```



```
enter initproc
>> sleep
[ERROR] [cpu0]: [sys_unmap]: file doesnt't exist!,Unmap failed!
BusyBox v1.33.1 (2022-09-26 07:33:21 UTC) multi-call binary.

Usage: sleep [N]...

Pause for a time equal to the total of the args given, where each arg can
have an optional suffix of (s)econds, (m)inutes, (h)ours, or (d)ays
>> cal
    January 1970
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

>> 
```

LOG

1. Oops

1. build

1. `error: failed to download`byteorder v1.4.3`

   Delete "--offline" in Makefile.

2. unstable特性

```
error[E0658]: `let...else` statements are unstable
   --> /home/ccyd/.cargo/git/checkouts/virtio-drivers-
4fdfaa862bcdc399/8e52ada/src/device/socket/vsock.rs:373:13
    |
373 | /             let Some(header) = self.pop_packet_from_rx_queue(body)?
else{
374 | |                 return Ok(None);
375 | |             };
    | |______________^
    |
    = note: see issue #87335 <https://github.com/rust-lang/rust/issues/87335>
for more information
    = help: add `#![feature(let_else)]` to the crate attributes to enable
......
......
error[E0658]: use of unstable library feature 'slice_ptr_len'
   --> /home/ccyd/.cargo/git/checkouts/virtio-drivers-
4fdfaa862bcdc399/8e52ada/src/transport/pci.rs:322:46
    |
322 |             if size_of::<T>() > config_space.len() * size_of::<u32>() {
    |                                                ^^^
    |
    = note: see issue #71146 <https://github.com/rust-lang/rust/issues/71146>
for more information
    = help: add `#![feature(slice_ptr_len)]` to the crate attributes to enable
```

试了很多，包括手动在报错的第三方库中添加宏，最后发现更新rust版本就解决了。

```
rustc 1.70.0-nightly (84dd17b56 2023-04-14)
```

3. `virtio_drivers`引用路径出错

```
error[E0432]: unresolved imports `virtio_drivers::VirtIOBlk`,
`virtio_drivers::VirtIOHeader`
 --> src/driver/block_device/virtio_blk.rs:1:22
  |
1 | use virtio_drivers::{VirtIOBlk, VirtIOHeader};
  |                      ^^^^^^^^^  ^^^^^^^^^^^^ no `VirtIOHeader` in the root
  |                      |
  |                      no `VirtIOBlk` in the root
  |
  = help: consider importing this struct instead:
          virtio_drivers::device::blk::VirtIOBlk
  = help: consider importing this struct instead:
          virtio_drivers::transport::mmio::VirtIOHeader
```

检查了库的修改历史，发现在一次commit后改变了框架结构。

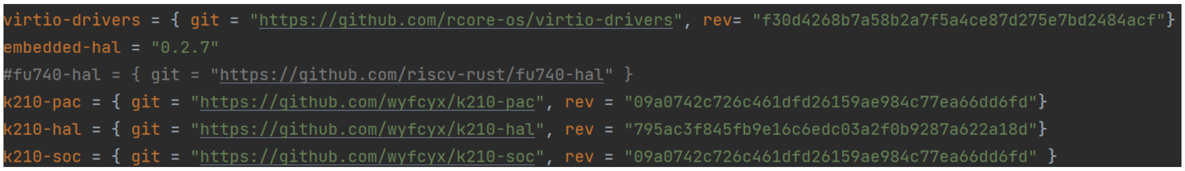
尝试了仓库提供的所有branch和tag，都没法正常运行，报错也各不相同。于是只好在最早的发行版"0.1.0"之前的commit里尝试，最终发现了`Oct 15, 2021`的commit版本可用，于是修改Cargo.toml:

```
virtio-drivers = { git = "https://github.com/rcore-os/virtio-drivers",rev =
"2aaf7d60c557ffdfaa2403bef420d41e469009ba"}
```

4. `k210-soc`出错

   同上，是由于仓库发生了变化。

   考虑到其他库也有可能，查找了`Cargo.toml`的git历史，找到每条依赖添加进来的时间，在各自的github仓库中找到最近一次的commit，手动在toml文件中为它们各自指定。

```
virtio-drivers = { git = "https://github.com/rcore-os/virtio-drivers", rev= "f30d4268b7a58b2a7f5a4ce87d275e7bd2484acf"}
embedded-hal = "0.2.7"
#fu740-hal = { git = "https://github.com/riscv-rust/fu740-hal" }
k210-pac = { git = "https://github.com/wyfcyx/k210-pac", rev = "09a0742c726c461dfd26159ae984c77ea66dd6fd"}
k210-hal = { git = "https://github.com/wyfcyx/k210-hal", rev = "795ac3f845fb9e16c6edc03a2f0b9287a622a18d"}
k210-soc = { git = "https://github.com/wyfcyx/k210-soc", rev = "09a0742c726c461dfd26159ae984c77ea66dd6fd" }
```

5. `k210-pac`出错

   删掉为它指定的commit就行了。。

6. `reference to packed field is unaligned`

```
error[E0793]: reference to packed field is unaligned
  --> src/fs/fat32/mod.rs:89:9
   |
89 |         assert_eq!(ebpb.root_dir_cluster, 2);
   |         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
   |
   = note: fields of packed structs are not properly aligned, and creating a
misaligned reference is undefined behavior (even if that reference is never
dereferenced)
   = help: copy the field contents to a local variable, or replace the reference
with a raw pointer and use `read_unaligned`/`write_unaligned` (loads and stores
via `*p` must be properly aligned even when using raw pointers)
   = note: this error originates in the macro `assert_eq` (in Nightly builds,
run with -Z macro-backtrace for more info)

error[E0793]: reference to packed field is unaligned
  --> src/fs/fat32/mod.rs:90:9
   |
90 |         assert_eq!(bpb.bytes_per_sector, 512);
   |         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
   |
   = note: fields of packed structs are not properly aligned, and creating a
misaligned reference is undefined behavior (even if that reference is never
dereferenced)
   = help: copy the field contents to a local variable, or replace the reference
with a raw pointer and use `read_unaligned`/`write_unaligned` (loads and stores
via `*p` must be properly aligned even when using raw pointers)
   = note: this error originates in the macro `assert_eq` (in Nightly builds,
run with -Z macro-backtrace for more info)
```

   这个错误完全搞不明白，尝试了很久，想到了前面的`unstable`特性问题，于是更换了几个rust的版本，发现每个版本都有不同报错，所以这个问题与rust版本有关。

按commit历史找了离Oops初始化最近的rust版本，修改后问题解决。

```
rustup install nightly-2022-04-11
rustup override set nightly-2022-04-11-x86_64-unknown-linux-gnu
rustup target add riscv64imac-unknown-none-elf --toolchain nightly-2022-04-11
rustup component add llvm-tools-preview --toolchain nightly-2022-04-11
```

2. run

1. 缺少fat32.img

   ```
   qemu-system-riscv64: -drive file=../fat32.img,if=none,format=raw,id=x0: Could not open '../fat32.img': No such file or directory
   ```

   在训练营stage2仓库找到了fat32镜像的下载地址。

2. `rust-objcopy`生成`kernel.bin`时卡死

   ```
   rust-objcopy --binary-architecture=riscv64 target/riscv64gc-unknown-none-elf/release/kernel --strip-all -O binary target/riscv64gc-unknown-none-elf/release/kernel.bin
   ```

   检查很久没有发现问题，`kernel.bin`已经生成了，用stat等工具大致检查过没发现问题。

   想到可能是卡在这一步之后，于是修改makefile:

   ```
   $(KERNEL_BIN): kernel_elf
       $(OBJCOPY) $(KERNEL_ELF) --strip-all -O binary $@
   ->
   $(KERNEL_BIN): kernel_elf
       echo "None"
   ```

   一样会卡。那么可以确定是卡在

   ```
   run: build
       @$(QEMU) $(QEMU-ARGS)
   ```

   的`@$(QEMU) $(QEMU-ARGS)`这一部，根据两个参数拼一下运行的语句：

   ```
   qemu-system-riscv64
   -machine $(QEMU_MACHINE)
   -smp $(CPUS)
   -nographic
   -bios $(BOOTLOADER)
   -device loader,file=$(KERNEL_BIN),addr=$(KERNEL_ENTRY)
   -drive file=$(FS_IMG_COPY),if=none,format=raw,id=x0
   -device virtio-blk-device,drive=x0,bus=virtio-mmio-bus.0
   ```

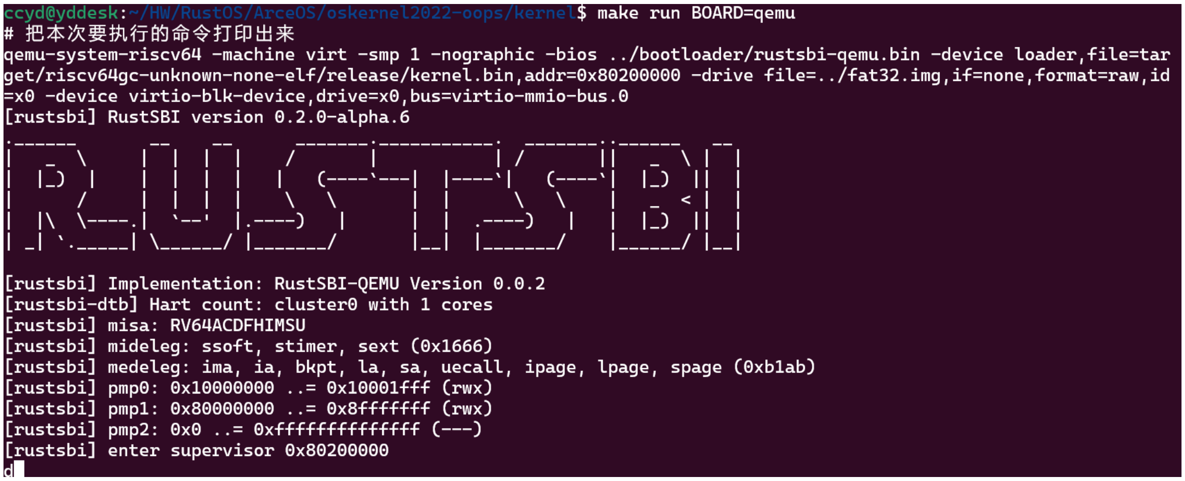
   感觉没问题，echo一下:

```
qemu-system-riscv64
-machine virt
-smp 1
-nographic
-bios ../bootloader/rustsbi-k210.bin
-device loader,file=target/riscv64gc-unknown-none-
elf/release/kernel.bin,addr=0x80200000
-drive file=../fat32.img,if=none,format=raw,id=x0
-device virtio-blk-device,drive=x0,bus=virtio-mmio-bus.0
```

发现虽然指定了目标平台为qemu，但`-bios ../bootloader/rustsbi-k210.bin`依然是k210开发板的sbi，可能是开发的同学在后期开发的过程中只考虑了k210板（于是在网上购买了一块板子），于是修改了Makefile:

```
BOOTLOADER  := ../bootloader/rustsbi-k210.bin
->
BOOTLOADER  := ../bootloader/rustsbi-$(BOARD).bin
```

成功运行:



```
ccyd@yddesk:~/HW/RustOS/ArceOS/oskernel2022-oops/kernel$ make run BOARD=qemu
# 把本次要执行的命令打印出来
qemu-system-riscv64 -machine virt -smp 1 -nographic -bios ../bootloader/rustsbi-qemu.bin -device loader,file=tar
get/riscv64gc-unknown-none-elf/release/kernel.bin,addr=0x80200000 -drive file=../fat32.img,if=none,format=raw,id
=x0 -device virtio-blk-device,drive=x0,bus=virtio-mmio-bus.0
[rustsbi] RustSBI version 0.2.0-alpha.6
.______       __    __      _______.___________.  _______..______   __
|   _  \     |  |  |  |    /       |           | /       ||   _  \ |  |
|  |_)  |    |  |  |  |   |   (----`---|  |----`|   (----`|  |_)  ||  |
|      /     |  |  |  |    \   \       |  |      \   \    |   _  < |  |
|  |\  \----.|  `--'  |.----)   |      |  |  .----)   |   |  |_)  ||  |
| _| `._____| \______/ |_______/       |__|  |_______/    |______/ |__|

[rustsbi] Implementation: RustSBI-QEMU Version 0.0.2
[rustsbi-dtb] Hart count: cluster0 with 1 cores
[rustsbi] misa: RV64ACDFHIMSU
[rustsbi] mideleg: ssoft, stimer, sext (0x1666)
[rustsbi] medeleg: ima, ia, bkpt, la, sa, uecall, ipage, lpage, spage (0xb1ab)
[rustsbi] pmp0: 0x10000000 ..= 0x10001fff (rwx)
[rustsbi] pmp1: 0x80000000 ..= 0x8fffffff (rwx)
[rustsbi] pmp2: 0x0 ..= 0xffffffffffffff (---)
[rustsbi] enter supervisor 0x80200000
d
```

2. testsuit

1. 编译toolchain

下载需要给wsl2设置代理，即使改用国内镜像它的子模块仍要通过github下载。

下载和编译的过程都很耗时间。

2. 编译`riscv-linux-rootfs`

1.

```
/home/ccyd/Env_For_Comp/riscv/sysroot/usr/include/gnu/stubs.h:14:11: fatal error:
gnu/stubs-lp64.h: No such file or directory
   14 | # include <gnu/stubs-lp64.h>
      |           ^~~~~~~~~~~~~~~~~~
compilation terminated.
```

解决办法:

```
cd /home/ccyd/Env_For_Comp/riscv/sysroot/usr/include/gnu/
cp stubs-lp64d.h stubs-lp64.h
```

2.

编译时报错xxx不存在，检查发现是WSL的PATH中来自windows的环境变量。

取消`appendWindowsPath`，在`vim /etc/wsl.conf`添加

```
# 不加载Windows中的PATH内容
[interop]
appendWindowsPath = false
```

3.

```
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

mkfs.fat 4.1 (2017-01-24)
*** failed to create riscv64-rootfs.bin
```

最后一步无法生成`riscv64-rootfs.bin`

TODO

3. fat32 fs info

> 附: **Linux系统中格式化sdcard为fat32文件系统**
>
> 1. 通过读卡器把sdcard插入Linux系统机器;
> 2. 卸载可能已有的sdcard分区;
>
> ```
> umount /dev/sdx*
> ```
>
> 1. 使用工具fdisk操作sdcard的盘号：/dev/sdx，盘号会根据实际机器不同而改变:

> ```
> fdisk /dev/sdx
> o #创建分区表
> n #新建分区
> p #设置为主分区
> 1 #设置为分区1
>   #后一路回车确认
>
> p #查看已创建的分区，这应该会显示有/dev/sdx1
>
> t #修改分区类型，选择分区1
> c #设置为fat32类型
>
> w #最后把修改写入sdcard盘，后退出
> ```
>
> 1. 执行格式化sdcard的分区
>
> ```
> mkfs.vfat -F 32 /dev/sdx1
> ```
>
> 1. 最后可查看sdcard盘的分区信息
>
> ```
> fdisk -l /dev/sdx
> ```

1. u盘挂载到wsl2

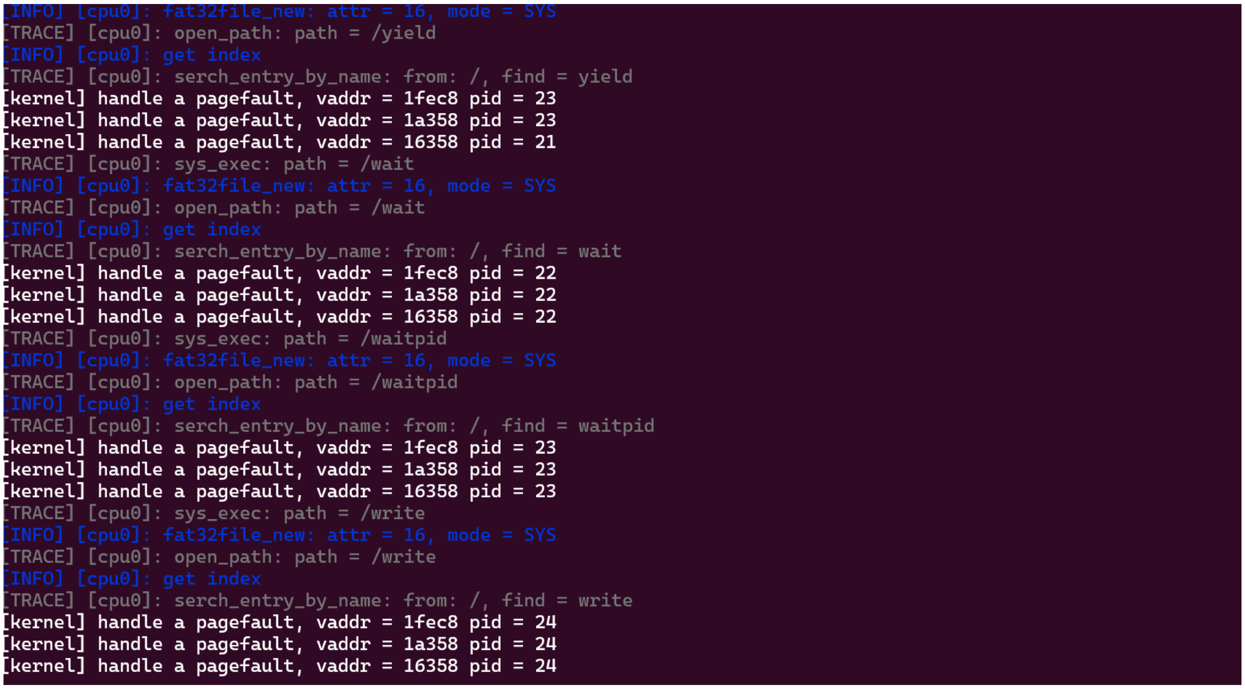
WSL本身并不支持连接 USB 设备，因此需要安装开源 usbipd-win 项目。

https://blog.csdn.net/qq_59475883/article/details/123299689

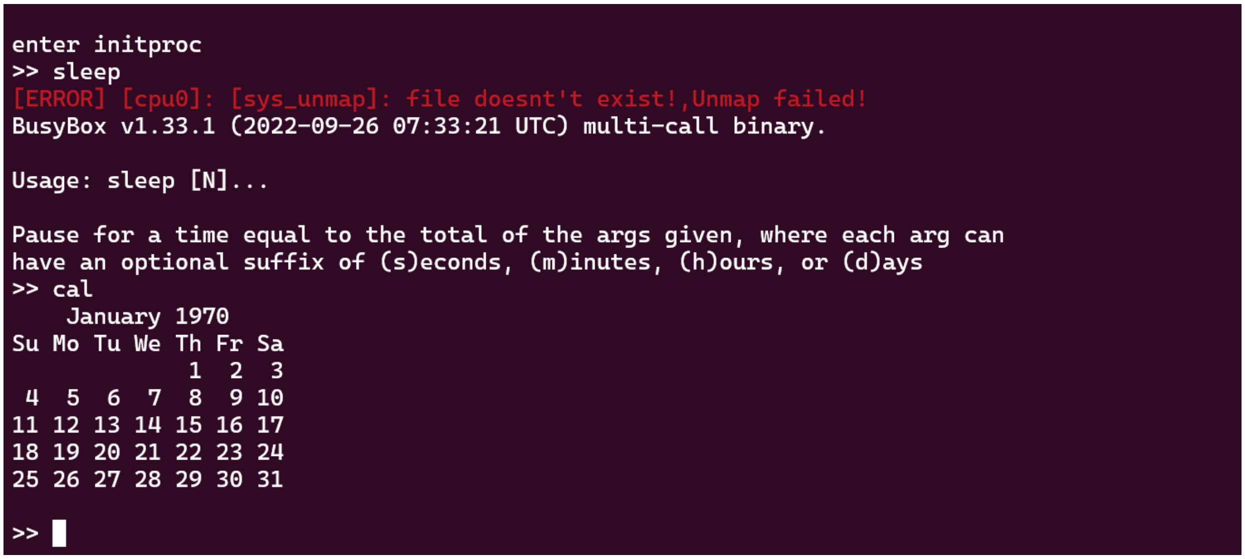
2. 改为本地挂载

```
dd if=/dev/zero of=disk.img bs=3M count=1024
mkfs.vfat -F 32 disk.img
sudo mount disk.img mnt
sudo cp -r ./src/* ./mnt/
sudo umount mnt
```

挂载后放在Oops运行



```
[INFO] [cpu0]: fat32file_new: attr = 16, mode = SYS
[TRACE] [cpu0]: open_path: path = /yield
[INFO] [cpu0]: get index
[TRACE] [cpu0]: serch_entry_by_name: from: /, find = yield
[kernel] handle a pagefault, vaddr = 1fec8 pid = 23
[kernel] handle a pagefault, vaddr = 1a358 pid = 23
[kernel] handle a pagefault, vaddr = 16358 pid = 21
[TRACE] [cpu0]: sys_exec: path = /wait
[INFO] [cpu0]: fat32file_new: attr = 16, mode = SYS
[TRACE] [cpu0]: open_path: path = /wait
[INFO] [cpu0]: get index
[TRACE] [cpu0]: serch_entry_by_name: from: /, find = wait
[kernel] handle a pagefault, vaddr = 1fec8 pid = 22
[kernel] handle a pagefault, vaddr = 1a358 pid = 22
[kernel] handle a pagefault, vaddr = 16358 pid = 22
[TRACE] [cpu0]: sys_exec: path = /waitpid
[INFO] [cpu0]: fat32file_new: attr = 16, mode = SYS
[TRACE] [cpu0]: open_path: path = /waitpid
[INFO] [cpu0]: get index
[TRACE] [cpu0]: serch_entry_by_name: from: /, find = waitpid
[kernel] handle a pagefault, vaddr = 1fec8 pid = 23
[kernel] handle a pagefault, vaddr = 1a358 pid = 23
[kernel] handle a pagefault, vaddr = 16358 pid = 23
[TRACE] [cpu0]: sys_exec: path = /write
[INFO] [cpu0]: fat32file_new: attr = 16, mode = SYS
[TRACE] [cpu0]: open_path: path = /write
[INFO] [cpu0]: get index
[TRACE] [cpu0]: serch_entry_by_name: from: /, find = write
[kernel] handle a pagefault, vaddr = 1fec8 pid = 24
[kernel] handle a pagefault, vaddr = 1a358 pid = 24
[kernel] handle a pagefault, vaddr = 16358 pid = 24
```



```
enter initproc
>> sleep
[ERROR] [cpu0]: [sys_unmap]: file doesnt't exist!,Unmap failed!
BusyBox v1.33.1 (2022-09-26 07:33:21 UTC) multi-call binary.

Usage: sleep [N]...

Pause for a time equal to the total of the args given, where each arg can
have an optional suffix of (s)econds, (m)inutes, (h)ours, or (d)ays
>> cal
    January 1970
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

>> 
```