Chair of Cyber-Physical Systems in Production Engineering
TUM School of Engineering and Design
Technical University of Munich

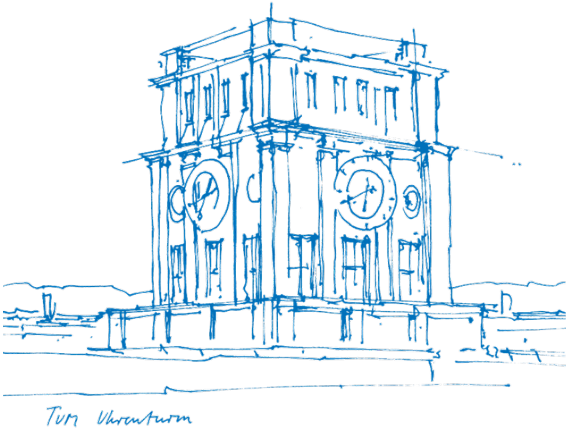
Jailhouse Workshop

Initial Build and Configuration

Wei, Zhihang

Chair of Cyber-Physical Systems in Production Engineering
TUM School of Engineering and Design
Technical University of Munich

Jun. 26th, 2024



Outline

What are needed?

For running Linux on ARM64:

- Linux Kernel
- Device Tree (.dts/.dtb)
- File System

Outline

Build Kernel

use existing config file

```
# make sure .config file exist
ls -lha .config

# run oldconfig with cross-compile options
make oldconfig ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu-

# build the kernel
make -j$(nproc) ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu-
```

Build Kernel
manually config something

```
# make sure .config file exist
ls -lha .config

# use menuconfig
make menuconfig ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu-

# build the kernel
make -j$(nproc) ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu-
```

Device Tree

.dtb and .dtc

```
# install dtc
sudo apt-get install device-tree-compiler

# convert a DTB file to a DTS file
dtc -I dtb -O dts -o output.dts input.dtb

# convert a DTS file back to a DTB file
dtc -I dts -O dtb -o output.dtb input.dts
```

File System

- dtc converter
- reserve space will be explained later

File System

- Do not forget git LFS

Put Them on Frodo/Sam

tftpboot for kernel and dtb

- link at /nftpboot/device/
- symbolic link to kernel and dtb

nfsroot for file system

- directory at /nfsroot

Outline

Configure Jailhouse

leave space for jailhouse hypervisor

In .dts:

```
reserved-memory {
    jailhouse_mem: jailhouse_mem@87d000000 {
            no-map;
            reg = <0x8 0x7d000000 0x0 0x03000000>;
    };
```

In e.g. configs/arm64/zynqmp-zcu102.c:

```
struct {
    .revision = JAILHOUSE_CONFIG_REVISION,
    .flags = JAILHOUSE_SYS_VIRTUAL_DEBUG_CONSOLE,
    .hypervisor_memory = {
            .phys_start = 0x87f000000,
            .size =       0x01000000,
    },
```

Build Jailhouse

```
# Jailhouse quick compile
make ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- \
    KDIR=/home/project/xilinx/linux-xilinx

# Jalihouse quick clean
make ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu-  \
    KDIR=/home/project/xilinx/linux-xilinx clean

# jailhouse module install
make ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- \
    INSTALL_MOD_PATH=/home/project/zcu-102-rootfs/rootfs \
    modules_install

# sync jailhouse files with FS (script provided by Andrea)
./sync_local_rootfs_jailhouse.sh
```

Sync FS

```
# sync local FS files with nfsroot on frodo/sam
# (script provided by Andrea)
./sync_frodo.sh
```

Outline

Start Jailhouse

root cell

```
cd /lib/modules/$(uname -r)/extra/driver/
insmod ./jailhouse.ko
cd ~
jailhouse enable jailhouse/zynqmp-zcu102.cell
```

Start Jailhouse

inmate cell: linux

```
jailhouse cell linux  \
    inmate_conf.cell \
    kernel_image \
    -d xx.dtb \
    -i initramfs \
    -c "console=ttyPS0,115200 root=/dev/ram0 rw"
```