EMBEDDED DEVICE DRIVERS

Linux Device Drivers on Beaglebone Black

Embedded Linux

- An Embedded Linux project involves
  - Obtaining, customizing and deploying 4 elements
    - Toolchain
      - Compiler and tools needed to create binaries from source
    - Bootloader
      - Pre-boot initialization of system and handover to OS
    - **Kernel**
      - **The heart of the system, managing resources and hardware**
    - Root file-system
      - Contains libraries and programs run after the kernel “boots”
      - Also holds the kernel modules

Course Focus: Kernel, Drivers

- This course focusses on:
  - Linux kernel
  - Device Drivers
- We start with
  - Creating an environment for kernel development
  - Involves:
    - Installing a compatible **toolchain** and **utils**
    - Setting up **kernel source tree**
    - Compiling kernel
    - Compiling modules

Embedded Linux: Toolchain

- We are using
  - GNU toolchain
    - For ARM EABI
    - with HF support
  - CROSS_COMPILE prefix
    - arm-linux-gnueabihf-

- Install this on Ubuntu using

  $ sudo apt-get install gcc-arm-linux-eabihf

Ubuntu: Utils needed

- For kernel configuration and compilation
  - build-essentials (for compilation of src code)
    $ sudo apt-get install build-essential git libmpc-dev
  - ncurses, bison, flex (for menuconfig)
    $ sudo apt-get install ncurses5-dev bison flex gettext
  - lz4 (for compression)
    $ sudo apt-get install lz4 lzop lzma
  - libssl-dev for ssh headers
    $ sudo apt-get install libssl-dev sshfs
  - u-boot-tools (for mkimage)
    $ sudo apt-get install u-boot-tools

Embedded Linux: Kernel

- Linux kernel – popular open-source project
  - Generic kernel: https://kernel.org
- SCM using ***git***
- Kernel versioning
  - New kernel version every 8-12 weeks
  - Refer this link for detailed explanation by ***greg-kh***
  - Schema
    - *<Major number>.<Minor number>.<Subminor number>*
      - *Example: 5.19.8*
    - *Followed by "localversion"*
      - *User created strings to differentiate*

Kernel, module compile for BBB

- BBB kernel: https://github.com/beagleboard/linux.git
- Commands:

  *$ git clone https://github.com/beagleboard/linux.git -b **v5.10.168-ti-r72** --depth=1*
  *$ cd **linux***

  *$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- **bb.org_defconfig***

  # Compile Linux kernel for BBB
  *$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- **uImage dtbs** LOADADDR=0x82000000 -j`nproc`*

  # Compile modules for BBB
  *$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- **modules** -j `nproc`*

- Clean up if you made a mistake

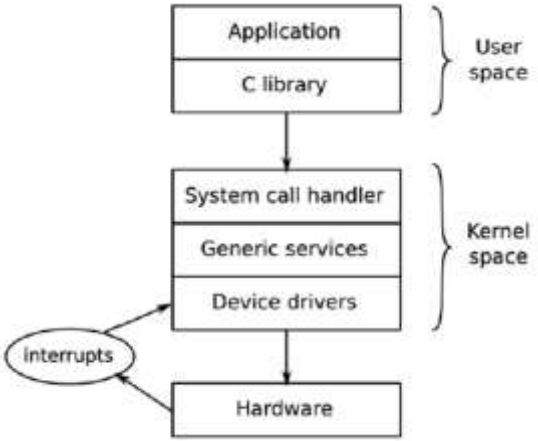
  *$ make mrproper # **cleaning up gracefully***

The kernel sources

- Directories in the kernel source tree
  - arch – *CPU architecture specific*
  - Documentation
  - drivers – *device drivers for hardware*
  - fs - *filesystems*
  - include – *include headers for kernel*
  - init – *initialization code*
  - kernel – *kernel proper, scheduler, lock, timers, etc.*
  - mm – *memory management*
  - net – *network stack*
  - scripts – *dtc scripts, etc.*
  - tools – *perf tools, etc.*

The role of the kernel

- Kernel roles
  - Initialize system, control hardware, handle interrupts
  - Operates in **kernel space**
- Application
  - Perform user-defined tasks
  - Operates in **user space / userland**
- Kernel space – User space
  - Bridged by
    - C library, System call interface



THANK YOU!