

# Building the Linux Kernel

**Advanced Embedded Software  
Development**  
with **Dan Walkes**



University of Colorado **Boulder**

**Learning objectives:**

**Introduction to KConfig**

**Kernel defconfig**

**Building the kernel**

**QEMU Example Kernel Build  
(assignment 3)**

# KConfig

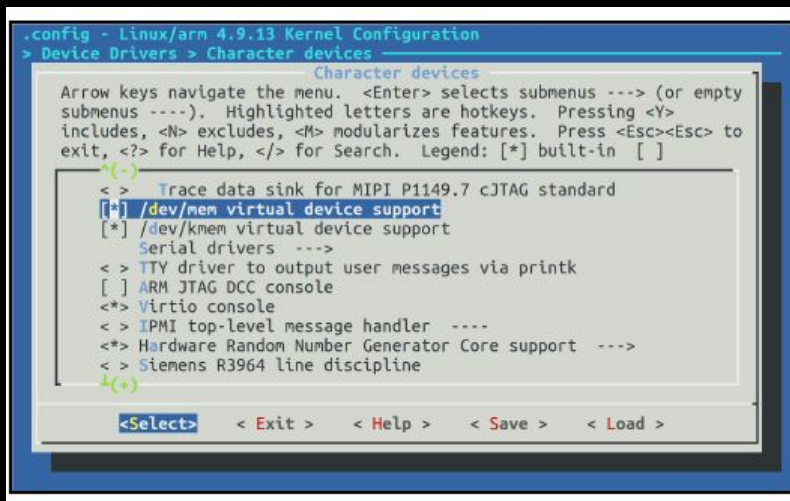
- Text format which controls kernel configuration
- drivers/char/Kconfig example

```
menu "Character devices"
[...]
config DEVMEM
    bool "/dev/mem virtual device support"
    default y
    help
        Say Y here if you want to support the /dev/mem device.
        The /dev/mem device is used to access areas of physical
        memory.
        When in doubt, say "Y".
[...]
endmenu
```

# Menuconfig

- Terminal based graphical tool to modify kernel configuration using KConfig files
- Invoke with “make menuconfig”

```
menu "Character devices"
[...]
config DEVMEM
    bool "/dev/mem virtual device support"
    default y
    help
        Say Y here if you want to support the /dev/mem device.
        The /dev/mem device is used to access areas of physical
        memory.
        When in doubt, say "Y".
[...]
endmenu
```



# KConfig

- Changes are stored in a .config file at kernel source root

```
menu "Character devices"
[...]
config DEVMEM
    bool "/dev/mem virtual device support"
    default y
```

```
CONFIG_VT_HW_CONSOLE_BINDING=y
CONFIG_UNIX98_PTYS=y
CONFIG_LEGACY_PTYS=y
CONFIG_LEGACY_PTY_COUNT=16
# CONFIG_SERIAL_NONSTANDARD is not set
# CONFIG_N_GSM is not set
# CONFIG_TRACE_SINK is not set
CONFIG_LDISC_AUTOLOAD=y
CONFIG_DEVMEM=y
# CONFIG_DEVMEM is not set
```

```
ecen5013@ecen5013-VirtualBox:~/assignment-3-work/linux-stable$ ls
arch      CREDITS      fs           Kbuild      LICENSES     modules.builtin  README      sound      virt
block     crypto       include      Kconfig     MAINTAINERS  modules.order    samples     System.map  vmlinux
certs     Documentation  init        kernel      Makefile     Module.symvers   scripts     tools       vmlinux.o
COPYING   drivers      ipc         lib         mm           net              security    usr
ecen5013@ecen5013-VirtualBox:~/assignment-3-work/linux-stable$ ls .config
.config
```

# KConfig Options

- Kconfig/menuconfig allows you to specify 3 modes for most drivers
  - off
    - Not built/included
  - on
    - built in to the kernel
  - Module
    - loadable at runtime

```
Misc devices --->
SCSI device support --->
< > Serial ATA and Parallel ATA drivers (libata) ----
[ ] Multiple devices driver support (RAID and LVM) ----
```

```
Misc devices --->
SCSI device support --->
< * > Serial ATA and Parallel ATA drivers (libata) ----
[ ] Multiple devices driver support (RAID and LVM) ----
```

```
Misc devices --->
SCSI device support --->
< M > Serial ATA and Parallel ATA drivers (libata) ----
[ ] Multiple devices driver support (RAID and LVM) ----
```

# Kernel Defconfigs

- Files which setup your .config file to use a set of configurations known to work for a specific hardware platform.
- Typically provided by hardware manufacturers (target development board)
- You can use these as the basis for your kernel .config

# Kernel Defconfigs



- What do we use for QEMU?
  - Assignment 3 uses QEMU virt
  - Can use defconfig with no arguments to generate
- See more options at <https://github.com/torvalds/linux/tree/master/arch/arm/configs>



# Compiling with Kbuild

- Example from drivers/char/Makefile
- `obj-y` = “built in”
- `obj-m` = “Module”
- needs `mem.o` and `random.o` built into the kernel (`-y`)
- `ttyprintk` may be included into the kernel or as a module based on `CONFIG_TTY_PRINTK`

```
obj-y                               += mem.o random.o
obj-$(CONFIG_TTY_PRINTK)           += ttyprintk.o
```

# Build Artifacts

- vmlinux
  - ELF binary - possibly with debugging symbols

# Build Artifacts

- Binaries in arch/\$ARCH/boot for use with bootloaders
  - Image - binary version of vmlinux
  - zImage - compressed Image
  - uImage - compressed image with U-Boot header

# Assignment 3 QEMU build - clean

- make ARCH=arm64  
CROSS\_COMPILE=aarch64-none-linux-gnu-  
**mrproper**
  - “deep clean” the kernel build tree -  
removing the .config file with any existing  
configurations

# Assignment 3 QEMU build - defconfig

- make ARCH=arm64  
CROSS\_COMPILE=aarch64-none-linux-gnu-  
defconfig
  - Configure for our “virt” arm dev board we  
will simulate in QEMU

# Assignment 3 QEMU build - vmlinux



- `make -j4 ARCH=arm64  
CROSS_COMPILE=aarch64-none-linux-gnu-  
all`
  - Build a kernel image for booting with  
QEMU

# Assignment 3 QEMU build - modules and devicetree

- make ARCH=arm64  
CROSS\_COMPILE=aarch64-none-linux-gnu-  
**modules**
  - Build any kernel modules
- make ARCH=arm64  
CROSS\_COMPILE=aarch64-none-linux-gnu- **dtbs**
  - Build the devicetree