Linux Kernel Training

Kernel building

Agenda

1. Get kernel sources
2. Configuring
3. Building kernel
4. Building rootfs
5. Launching

Get kernel sources

Get the latest stable kernel from https://www.kernel.org/:

- Clone the linux-stable repository and switch to the latest stable branch (v4.13)

```
git clone
git://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable
cd linux-stable
git checkout v4.13 -b work
```

- Alternatively you can create shallow clone

```
git clone
git://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable
-b v4.13 --depth 1
```

- Or download it as tarball

```
wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.13.tar.xz
tar -xJf linux-4.13.tar.xz
```

Configure and build

- Prepare build directory

```
export BUILD_KERNEL=<your_build_path>
make ARCH=i386 O=${BUILD_KERNEL} defconfig
cd ${BUILD_KERNEL}
```

- Configure kernel

```
make menuconfig
```

- Build the kernel

```
make (you can use -j4 to speedup building)
```

Built images are located in **`${BUILD_KERNEL}/arch/${ARCH}/boot/`**

Buildroot

https://buildroot.org/

- Get buildroot sources

```
git clone git://git.buildroot.net/buildroot
cd buildroot
```

- Prepare build directory

```
export BUILD_ROOTFS=<your_build_path>
make O=${BUILD_ROOTFS} qemu_x86_defconfig
cd ${BUILD_ROOTFS}
```

- Configure buildroot

```
make menuconfig
```

Buildroot configuration

- Target options:
  - Target Architecture = **i386**
  - Target Architecture Variant = **i686**
- Toolchain:
  - Custom kernel headers series = **4.13.x** *(should match the kernel)*
  - **[*]** Enable WCHAR support
- System configuration:
  - System hostname = **myLinux** *(give it a name)*
  - System banner = **Welcome to myLinux**
  - **[*]** Enable root login with password
  - Root password = **<*rootpass*>**
  - Path to the users tables = ***${BUILD_ROOTFS}*/users** *(we'll create regular user)*
  - Root filesystem overlay directories = ***${BUILD_ROOTFS}*/root** *(we'll put there some additional configs)*
- Kernel:
  - **[ ]** Linux Kernel *(we'll use manually built kernel, so disable it here)*

Buildroot configuration

- Target packages:
  - **[*]** Show packages that are also provided by busybox
  - Development tools
    - **[*]** binutils
    - **[*]** binutils binaries
    - **[*]** findutils
    - **[*]** grep
    - **[*]** sed
    - **[*]** tree
  - Libraries:
    - Compression and decompression:
      - **[*]** zlib
    - Text and terminal handling:
      - **[*]** ncurses
      - **[*]** readline
  - Networking applications:
    - **[*]** dropbear *(ssh server)*
    - [*] wget
  - Shell and utilities:
    - **[*]** bash
    - **[*]** file
    - **[*]** sudo
    - **[*]** which
  - System tools
    - **[*]** kmod
    - **[*]** kmod utilities
    - **[*]** rsyslog
  - Text editors and viewers:
    - [*] joe *(it might be the easiest terminal editor unless you're familiar with vi)*
    - **[*]** less
    - **[*]** mc
    - **[*]** vim
- Filesystem images:
  - **[*]** ext2/3/4 root filesystem
  - ext2/3/4 variant = **ext3**
  - **[*]** tar the root filesystem

Finish setup and build

- Additional files
  - Create user record

    ```
    echo "user 1000 user 1000 =pass /home/user /bin/bash - Linux
    User" > ${BUILD_ROOTFS}/users
    ```

    (This will create user user with id 1000 and password pass)
  - Add the user to sudoers

    ```
    mkdir -p ${BUILD_ROOTFS}/root/etc/sudoers.d
    echo "user     ALL=(ALL) ALL" > ${BUILD_ROOTFS}/root/etc/sudoers.d/user
    ```

  - Create list of shells for dropbear

    ```
    mkdir -p ${BUILD_ROOTFS}/root/etc
    echo "/bin/sh" > ${BUILD_ROOTFS}/root/etc/shells
    echo "/bin/bash" >> ${BUILD_ROOTFS}/root/etc/shells
    ```
- Build the FS

  ```
  make
  ```

  It will take some time, but eventually you'll get ${BUILD_ROOTFS}/images/rootfs.ext3

Launch

- Launch QUEMU

```
qemu-system-i386 \
-kernel ${BUILD_KERNEL}/arch/x86/boot/bzImage \
-append "root=/dev/sda" \
-hda ${BUILD_ROOTFS}/images/rootfs.ext3 \
-redir tcp:8022::22 &
```

Note: We are running qemu in background, but bound to current terminal.

- Login to your system using ssh

```
ssh -p 8022 user@localhost
ssh myLinux
```

- Tips

~/.ssh/config:

```
Host myLinux
    HostName
localhost
    Port     8022
    User     user
```

Let’s Go!