

Assignment 4: Buildroot Tips

**Advanced Embedded Linux
Development
with Dan Walkes**



University of Colorado **Boulder**

Learning objectives:

Additional tips for using Buildroot and
Completing Assignment 4

dirname \$0 trick

- \$0 is full execution name, like /path/to/script.sh if you start a script using ./path/to/script.sh
- dirname gives us /path/to
- cd `dirname \$0` means change to the directory where the script is located

Single Package Rebuild/Local Build



- Single Package Rebuild
 - Use `make aesc-assignments-rebuild` or `aesc-assignments-reconfigure` from the buildroot directory
- Local builds
 - Use argument `AESD_ASSIGNMENTS_OVERRIDE_SRCDIR=/path/to/local`
- These save time! Use them!
 - Don't do a clean build each time you make a change

GNU screen

- What happens if you start the build from a terminal session and your session dies or is closed?
 - Build stops
- Alternative: Setup a dedicated screen for your build
 - `screen -S buildroot-build`
 - start the build
 - `Ctrl+a d`
 - To reconnect, `screen -r buildroot-build`

ssh_agent

- Build needs to start by installing toolchain.
- Will be a while before it tries to compile your package
- Will stall there waiting for you to enter your SSH credentials for your key
- Instead use passwordless key or setup ssh-agent in your terminal session
 - `eval `ssh-agent``
 - `ssh-add ~/.ssh/yourprivatekey`

Avoid Spaces In Paths!



The bug [#712](#) suggests that make does not handle names with spaces. Nowhere, never.

49

I found a [blog post saying it's partially implemented](#) by escaping the spaces with `\` (`\\` seems to be typo or formatting artefact), but:



- It does not work in any functions except `$(wildcard)`.
- It does not work when expanding lists of names from variables, which includes the special variables `$?`, `$^` and `$+` as well as any user-defined variable. Which in turn means that while `$(wildcard)` will match correct files, you won't be able to interpret the result anyway.

So with explicit or very simple pattern rules you can get it to work, but beyond that you are out of luck. You'll have to look for some other build system that does support spaces. I am not sure whether [jam/bjam](#) does, [scons](#), [waf](#), [ant](#), [nant](#) and [msbuild](#) all should work.

QEMU ssh passthrough

- SSH - Secure Shell
- Secure remote command line access
- `ssh user@ip-or-hostname`
- Default SSH port is TCP port 22
- Our QEMU instance can't use this port (as it's in use by VM)
- Need to modify `runqemu` script to forward port 10022 to our QEMU instance using `hostfwd`
- Then connect `ssh -p 10022 root@localhost`
 - Use capital `-P` for `scp` command

Buildroot Full Rebuilds

- “it is the responsibility of the user to know when a full rebuild is necessary”
 - Target architecture/toolchain config changed “some other fundamental configuration item”
 - Package is removed
 - When package referencing other packages are added (in some cases)
 - Package sub-options are changed
- Bottom line - `./clean.sh && ./build.sh` is usually a troubleshooting step but don't use this for incremental changes to your package.
 - See single package rebuild/local build instructions

Buildroot Output Directory

- The output/target directory contains the “almost complete” content which will make up your root filesystem.
 - Good for sanity checking whether things are placed where you think they are placed.
 - Can delete files here which you moved/deleted in a package to avoid the need to full rebuild.

```
ecen5013@ecen5013-VirtualBox:~/buildroot-complete-private/buildroot/output$ ls target/  
bin  etc  lib64  media  opt  root  sbin  THIS_IS_NOT_YOUR_ROOT_FILESYSTEM  usr  
dev  lib  linuxrc  mnt  proc  run  sys  tmp  var
```