

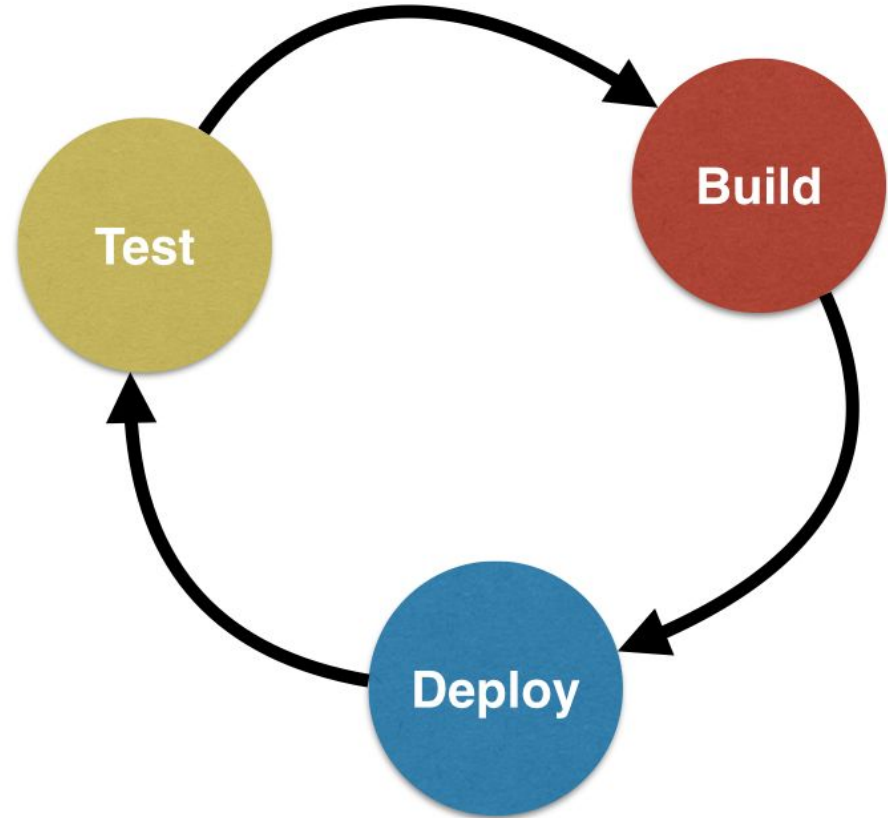
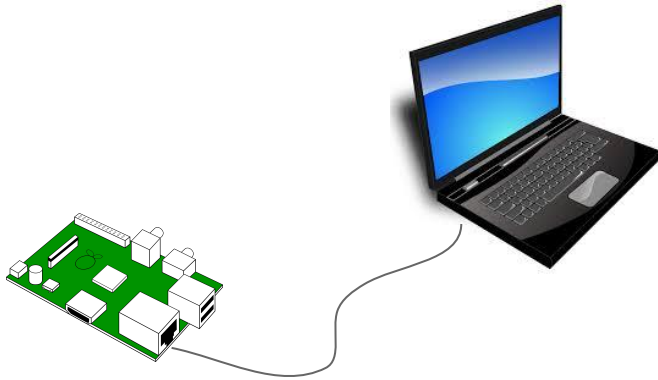


Deploy Software Updates for Linux Devices

Strategies for developing and deploying your embedded applications and images

Scope

- Development workflow
- Application
- Embedded systems
- What tools are available



Session overview

- Desktop environment
- Embedded environment
- Development workflow (simple)
 - Package managers
 - Yocto/OE-core Package Management
- Development workflow (advanced)
 - Network boot
 - OTA updater as developer tool



About me

- Mirza Krak

- 7 years in Embedded Linux development
- U-boot and Linux kernel development
- Yocto/Buildroot
- mirza.krak@northern.tech

- mender.io

- Over-the-air updater for Embedded Linux
- Open source (Apache License, v2)
- Dual A/B rootfs layout (client)
- Remote deployment management (server)
- Under active development



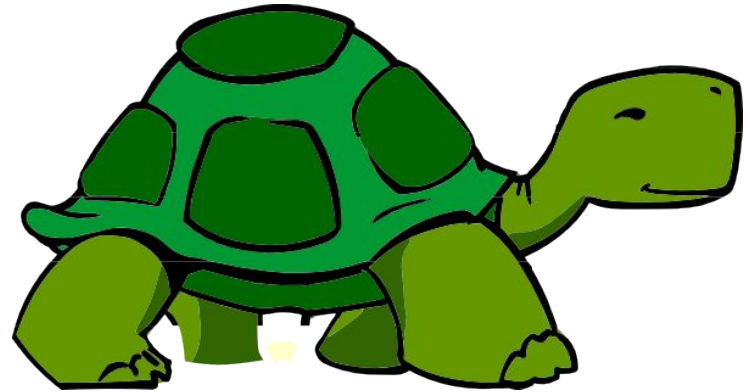
Desktop environment

- High availability
 - “apt-get install”
 - trace and debug tools
- Same machine
 - Build, Run, Test
- Short cycles
- Keep development here
 - “Mock” hardware
- Be aware



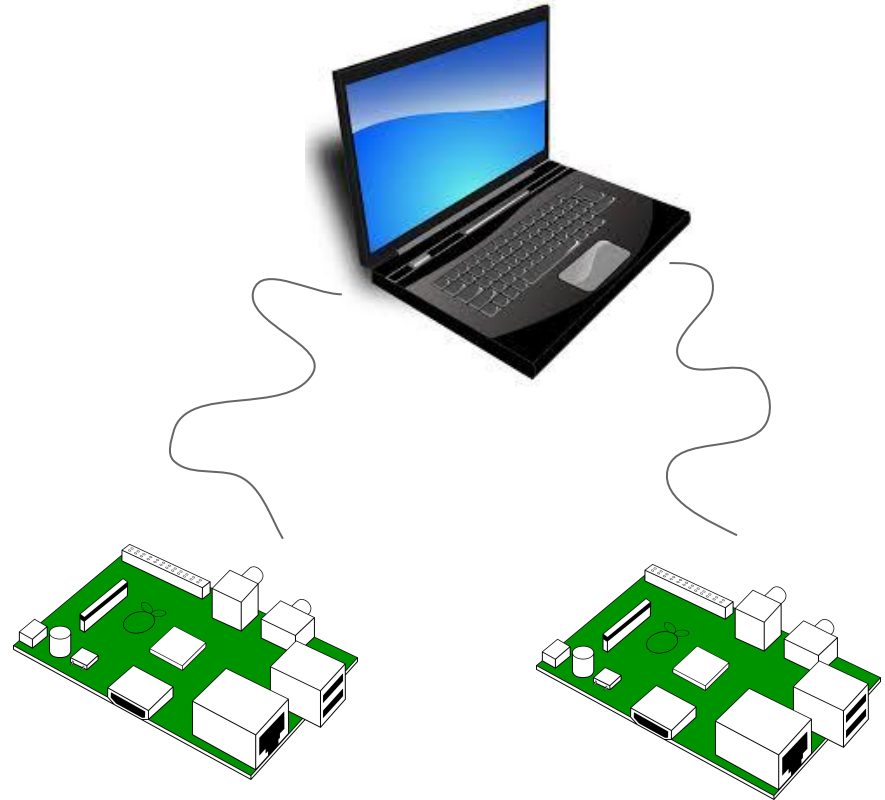
Desktop environment on embedded

- Possible
- Armbian/Ubuntu/Raspbian
- Drawbacks
 - slow compile times
 - slow in general compared to PC
 - IDE, favorite editors
 - Not really viable in the long run



Cross device development

- Cross-compile
- Accepted approach
- Introduces complexity
- Multiple devices



Transfer files

- Entry point
- Transfer files
 - Manual work
 - Error prone
 - Hard to replicate across many devices



```
scp application  
root@device:/usr/bin
```

```
tftp -g -r application  
hostname
```

```
scp *.conf root@device:/etc
```



Transfer files - IDE

- Eclipse, Qt Creator..
 - Cross-compile
 - Post-build hooks



eclipse



Package managers

- Package Manager
 - A collection of the software tools for automating the process of installing, upgrading, configuring and removing packages
- On target package managers
 - opkg, deb, rpm
- Common package management utilities/systems
 - apt, yum, dnf, pacman, zypper, smart



Package managers

- Package application + additional files
 - “make dpkg”
- More control
- More sanity checks
 - Dependency tracking
 - Upstream package feeds
 - Custom package feeds
- Useful during development
 - especially early phase of projects
 - Utilities (strace, evtest, tcpdump, iperf)
- Not always available



Package managers comparison

	deb	rpm	ipkg	opkg
File format	.deb	.rpm	.ipk	.ipk
License	GPL	GPL	GPL v2	GPL v2
Development status	active	active	discontinued	active
Yocto/OE support	yes	yes	no	yes
Package installation	dpkg -i file.deb	rpm -i file.rpm	ipkg install file.ipk	opkg install file.ipk
Build time (Yocto image)	~226 minutes	~251 minutes		~209 minutes



OE-core + Package managers

- The Angstrom Distribution
 - maintains opkg package feeds
 - meta-angstrom
 - angstrom-v2018.06-sumo
 - DISTRO = “angstrom”

“opkg update && opkg install vim”



Yocto/OE-core + Package managers

- Yocto/OE-core generates packages (rpm, deb, ipk)
 - `PACKAGE_CLASSES ?= "package_ipk"`
 - `build/tmp/deploy/ipk/`
- Easy to convert to a custom package feed
 - “bitbake package-index”
- Serve the feed on the network
 - “`python -m SimpleHTTPServer 8000`”



Yocto + Package managers

- Tools on target
 - `EXTRA_IMAGE_FEATURES += " package-management "`
- Configuration on target
 - `meta-openembedded/meta-oe`
 - `distro-feed-configs.bb`

- `DISTRO_FEED_PREFIX`
- `DISTRO_FEED_URI`
- `DISTRO_FEED_ARCHS`



Yocto + Package managers

```
.  
└─ etc  
  └─ opkg  
    ├── all-feed.conf  
    ├── core2-64-feed.conf  
    ├── corei7-64-feed.conf  
    ├── corei7-64-intel-common-feed.conf  
    ├── intel_corei7_64-feed.conf  
    └─ x86_64-feed.conf
```

```
$ cat etc/opkg/core2-64-feed.conf
```

```
src/gz remote-core2-64 http://my-distribution.example/remote-feed//core2-64
```



Yocto + Package managers

- Workflow
 - make changes
 - rebuild package index
 - “opkg update && opkg install” on device
- “bitbake world -k”



Configuration management tools

- Configuration management tool
 - A tool used for populating and enforcing host configuration (adding/removing/updating software, adding/removing users, changing files permissions, ...)
- Tools available
 - CFEngine, Puppet, Chef, Ansible
- Configuration strategy
 - Install the “golden image” on the device
 - Install a CM server and create a set of rules for managing/changing the device configuration
 - Setup connectivity and trust between CM server and the device
 - Change the device configuration using the CM agent



Scope bigger than a single binary

- Custom kernel options
- Customer configuration options on system applications
 - systemd, busybox, network manager
- Custom hardware



Network boot

- All resources are available on the network
- Some complexity involved in the setup
- Requires reboot on the target device to load new software
- Can be easily extended to multiple devices
- Common in CI/CD



Network boot

- PXELINUX
- tftp / pxe
 - Kernel (ulmage) and dtb
 - initrd

```
-----/tftpboot/pxelinux.cfg/menus/linux.list-----
```

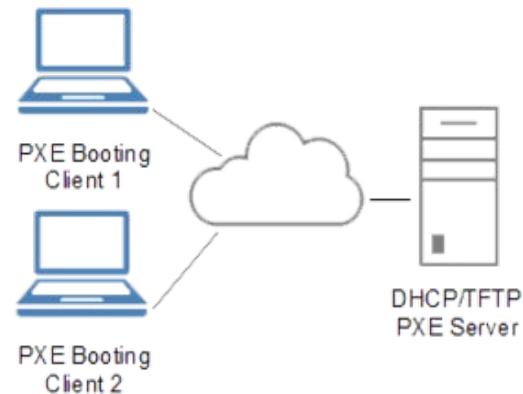
```
menu title Linux selections
```

```
# Just another label
```

```
label linux-2.6.38
```

```
kernel kernels/linux-2.6.38.bin
```

```
append root=/dev/sdb1
```



Network boot

- Script it
- NFS root file-system
- tftp/tftpboot

```
setenv ipaddr 192.168.1.3
```

```
setenv serverip 192.168.1.2
```

```
setenv bootargs 'ignore_loglevel rw root=/dev/nfs nfsroot=192.168.1.2:/nfs/h3ulcb,nfsvers=3  
ip=192.168.1.3:192.168.1.2::255.255.255.0:h3ulcb'
```

```
tftp 0x48080000 Image; tftp 0x48000000 Image-r8a7795-h3ulcb.dtb; booti 0x48080000 - 0x48000000
```

<https://elinux.org/R-Car/Boards/Yocto-Gen3>



Update solutions

- Can be used as a development tool
 - mender, rauc, swupdate...
- Integrate early in the development cycle
- Many benefits
 - Similar to production
 - Validation of the update solution
 - Image based update
 - Simplifies testing
 - Stateless
 - Avoid bricking devices



Update solutions

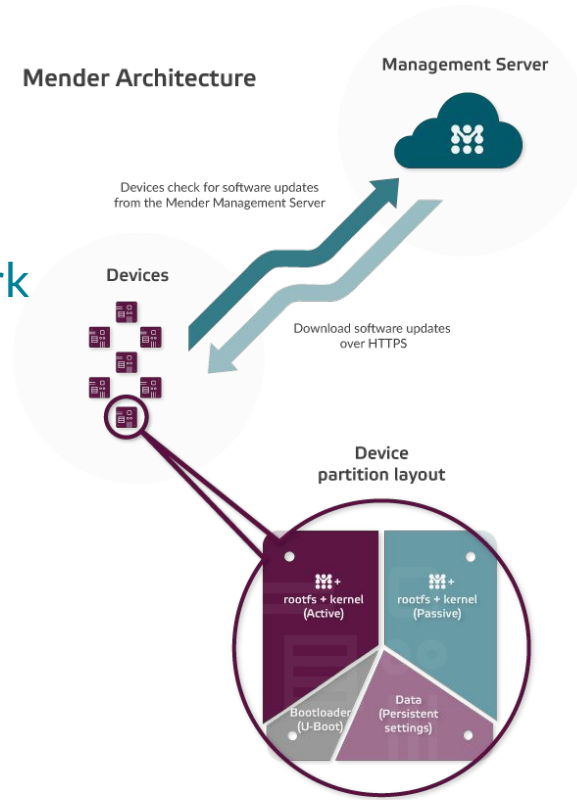
- Fits well into the developer workflow
 - Easy integration with CI/CD systems



OTA updater

- Mender
 - A/B image updates
- Standalone mode
 - CLI
 - Supports fetching updates via network
- OTA updates
 - Can be integrated with CI/CD

mender -rootfs <http://192.168.1.10/core-image-base.mender>



Thank you

Questions?

