سرفصل آموزشی

پک پایه مهندسی Embedded Linux

فهرست سرفصل‌های دوره‌های آموزشی

سرفصل‌های دوره آموزشی

LPIC-1

Topic 101: System Architecture

101.1 Determine and configure hardware settings

Weight: 2

Description: Candidates should be able to determine and configure fundamental system hardware

Key Knowledge Areas:

- Enable and disable integrated peripherals.
- Differentiate between the various types of mass storage devices.
- Determine hardware resources for devices.
- Tools and utilities to list various hardware information (e.g. lsusb, lspci, etc.).
- Tools and utilities to manipulate USB devices.
- Conceptual understanding of sysfs, udev and dbus.

The following is a partial list of the used files, terms and utilities:

- /sys/
- /proc/
- /dev/
- modprobe
- lsmod
- lspci
- lsusb

101.2 Boot the system

Weight: 3

Description: Candidates should be able to guide the system through the booting process.

Key Knowledge Areas:

- Provide common commands to the boot loader and options to the kernel at boot time.
- Demonstrate knowledge of the boot sequence from BIOS/UEFI to boot completion.
- Understanding of SysVinit and systemd.
- Awareness of Upstart.
- Check boot events in the log files.

The following is a partial list of the used files, terms and utilities:

- dmesg
- journalctl
- BIOS
- UEFI
- bootloader
- kernel
- initramfs
- init
- SysVinit
- systemd

101.3 Change runlevels / boot targets and shutdown or reboot system

Weight: 3

Description: Candidates should be able to manage the SysVinit runlevel or systemd boot target of the system. This objective includes changing to single user mode, shutdown or rebooting the system. Candidates should be able to alert users before switching runlevels / boot targets and properly terminate processes. This objective also includes setting the default SysVinit runlevel or systemd boot target. It also includes awareness of Upstart as an alternative to SysVinit or systemd.

Key Knowledge Areas:

- Set the default runlevel or boot target.
- Change between runlevels / boot targets including single user mode.
- Shutdown and reboot from the command line.
- Alert users before switching runlevels / boot targets or other major system events.
- Properly terminate processes.
- Awareness of acpid.

The following is a partial list of the used files, terms and utilities:

- /etc/inittab
- shutdown
- init
- /etc/init.d/
- telinit
- systemd
- systemctl
- /etc/systemd/
- /usr/lib/systemd/
- wall

Topic 102: Linux Installation and Package Management

102.1 Design hard disk layout

Weight: 2

Description: Candidates should be able to design a disk partitioning scheme for a Linux system.

Key Knowledge Areas:

- Allocate filesystems and swap space to separate partitions or disks.
- Tailor the design to the intended use of the system.
- Ensure the /boot partition conforms to the hardware architecture requirements for booting.
- Knowledge of basic features of LVM.

The following is a partial list of the used files, terms and utilities:

- / (root) filesystem
- /var filesystem
- /home filesystem
- /boot filesystem
- EFI System Partition (ESP)
- swap space
- mount points
- partitions

102.2 Install a boot manager

Weight: 2

Description: Candidates should be able to select, install and configure a boot manager.

Key Knowledge Areas:

- Providing alternative boot locations and backup boot options.
- Install and configure a boot loader such as GRUB Legacy.
- Perform basic configuration changes for GRUB 2.
- Interact with the boot loader.

The following is a partial list of the used files, terms and utilities:

- menu.lst, grub.cfg and grub.conf
- grub-install
- grub-mkconfig
- MBR

102.3 Manage shared libraries

Weight: 1

Description: Candidates should be able to determine the shared libraries that executable programs depend on and install them when necessary.

Key Knowledge Areas:

- Identify shared libraries.
- Identify the typical locations of system libraries.
- Load shared libraries.

The following is a partial list of the used files, terms and utilities:

- ldd
- ldconfig
- /etc/ld.so.conf
- LD_LIBRARY_PATH

102.4 Use Debian package management

Weight: 3

Description: Candidates should be able to perform package management using the Debian package tools.

Key Knowledge Areas:

- Install, upgrade and uninstall Debian binary packages.
- Find packages containing specific files or libraries which may or may not be installed.
- Obtain package information like version, content, dependencies, package integrity and installation status (whether or not the package is installed).
- Awareness of apt.

The following is a partial list of the used files, terms and utilities:

- /etc/apt/sources.list
- dpkg
- dpkg-reconfigure
- apt-get
- apt-cache

102.5 Use RPM and YUM package management

Weight 3

Description: Candidates should be able to perform package management using RPM, YUM and Zypper.

Key Knowledge Areas:

- Install, re-install, upgrade and remove packages using RPM, YUM and Zypper.
- Obtain information on RPM packages such as version, status, dependencies, integrity and signatures.
- Determine what files a package provides, as well as find which package a specific file comes from.
- Awareness of dnf.

The following is a partial list of the used files, terms and utilities:

- rpm
- rpm2cpio
- /etc/yum.conf
- /etc/yum.repos.d/
- yum
- zypper

102.6 Linux as a virtualization guest

Weight: 1

Description: Candidates should understand the implications of virtualization and cloud computing on a Linux guest system.

Key Knowledge Areas:

- Understand the general concept of virtual machines and containers.
- Understand common elements virtual machines in an IaaS cloud, such as computing instances, block storage and networking.
- Understand unique properties of a Linux system which have to changed when a system is cloned or used as a template.
- Understand how system images are used to deploy virtual machines, cloud instances and containers.
- Understand Linux extensions which integrate Linux with a virtualization product.
- Awareness of cloud-init.

The following is a partial list of the used files, terms and utilities:

- Virtual machine
- Linux container
- Application container
- Guest drivers
- SSH host keys
- D-Bus machine id

Topic 103: GNU and Unix Commands

103.1 Work on the command line

Weight: 4

Description: Candidates should be able to interact with shells and commands using the command line. The objective assumes the Bash shell.

Key Knowledge Areas:

- Use single shell commands and one line command sequences to perform basic tasks on the command line.
- Use and modify the shell environment including defining, referencing and exporting environment variables.
- Use and edit command history.
- Invoke commands inside and outside the defined path.

The following is a partial list of the used files, terms and utilities:

- bash
- echo
- env
- export
- pwd
- set
- unset
- type
- which
- man
- uname
- history
- .bash_history
- Quoting

103.2 Process text streams using filters

Weight: 2

Description: Candidates should be able to apply filters to text streams.

Key Knowledge Areas:

- Send text files and output streams through text utility filters to modify the output using standard UNIX commands found in the GNU textutils package.

The following is a partial list of the used files, terms and utilities:

- bzcat
- cat
- cut
- head
- less
- md5sum
- nl
- od
- paste
- sed
- sha256sum
- sha512sum
- sort
- split
- tail
- tr
- uniq
- wc
- xzcat
- zcat

103.3 Perform basic file management

Weight: 4

Description: Candidates should be able to use the basic Linux commands to manage files and directories.

Key Knowledge Areas:

- Copy, move and remove files and directories individually.
- Copy multiple files and directories recursively.
- Remove files and directories recursively.
- Use simple and advanced wildcard specifications in commands.
- Using find to locate and act on files based on type, size, or time.
- Usage of tar, cpio and dd.

The following is a partial list of the used files, terms and utilities:

- cp
- find
- mkdir
- mv
- ls
- rm
- rmdir
- touch
- tar
- cpio
- dd
- file
- gzip
- gunzip

- bzip2
- bunzip2
- xz
- unxz
- file globbing

103.4 Use streams, pipes and redirects

Weight: 4

Description: Candidates should be able to redirect streams and connect them in order to efficiently process textual data. Tasks include redirecting standard input, standard output and standard error, piping the output of one command to the input of another command, using the output of one command as arguments to another command and sending output to both stdout and a file.

Key Knowledge Areas:

- Redirecting standard input, standard output and standard error.
- Pipe the output of one command to the input of another command.
- Use the output of one command as arguments to another command.
- Send output to both stdout and a file.

The following is a partial list of the used files, terms and utilities:

- tee
- xargs

103.5 Create, monitor and kill processes

Weight: 4

Description: Candidates should be able to perform basic process management.

Key Knowledge Areas:

- Run jobs in the foreground and background.
- Signal a program to continue running after logout.
- Monitor active processes.
- Select and sort processes for display.
- Send signals to processes.

The following is a partial list of the used files, terms and utilities:

- &
- bg
- fg
- jobs
- kill
- nohup
- ps
- top
- free
- uptime
- pgrep
- pkill
- killall
- watch
- screen
- tmux

103.6 Modify process execution priorities

Weight: 2

Description: Candidates should should be able to manage process execution priorities.

Key Knowledge Areas:

- Know the default priority of a job that is created.
- Run a program with higher or lower priority than the default.
- Change the priority of a running process.

The following is a partial list of the used files, terms and utilities:

- nice
- ps
- renice
- top

103.7 Search text files using regular expressions

Weight: 3

Description: Candidates should be able to manipulate files and text data using regular expressions. This objective includes creating simple regular expressions containing several notational elements as well as understanding the differences between basic and extended regular expressions. It also includes using regular expression tools to perform searches through a filesystem or file content.

Key Knowledge Areas:

- Create simple regular expressions containing several notational elements.
- Understand the differences between basic and extended regular expressions.
- Understand the concepts of special characters, character classes, quantifiers and anchors.
- Use regular expression tools to perform searches through a filesystem or file content.
- Use regular expressions to delete, change and substitute text.

The following is a partial list of the used files, terms and utilities:

- grep
- egrep
- fgrep
- sed
- regex(7)

103.8 Basic file editing

Weight: 3

Description: Candidates should be able to edit text files using vi. This objective includes vi navigation, vi modes, inserting, editing, deleting, copying and finding text. It also includes awareness of other common editors and setting the default editor.

Key Knowledge Areas:

- Navigate a document using vi.
- Understand and use vi modes.
- Insert, edit, delete, copy and find text in vi.
- Awareness of Emacs, nano and vim.
- Configure the standard editor.

Terms and Utilities:

- vi
- /, ?
- h,j,k,l
- i, o, a
- d, p, y, dd, yy
- ZZ, :w!, :q!
- EDITOR

Topic 104: Devices, Linux Filesystems, Filesystem Hierarchy Standard

104.1 Create partitions and filesystems

Weight: 2

Description: Candidates should be able to configure disk partitions and then create filesystems on media such as hard disks. This includes the handling of swap partitions.

Key Knowledge Areas:

- Manage MBR and GPT partition tables
- Use various mkfs commands to create various filesystems such as:
- ext2/ext3/ext4
- XFS
- VFAT
- exFAT
- Basic feature knowledge of Btrfs, including multi-device filesystems, compression and subvolumes.

The following is a partial list of the used files, terms and utilities:

- fdisk
- gdisk
- parted
- mkfs
- mkswap

104.2 Maintain the integrity of filesystems

Weight: 2

Description: Candidates should be able to maintain a standard filesystem, as well as the extra data associated with a journaling filesystem.

Key Knowledge Areas:

- Verify the integrity of filesystems.
- Monitor free space and inodes.
- Repair simple filesystem problems.

The following is a partial list of the used files, terms and utilities:

- du
- df
- fsck
- e2fsck
- mke2fs
- tune2fs
- xfs_repair
- xfs_fsr
- xfs_db

104.3 Control mounting and unmounting of filesystems

Weight: 3

Description: Candidates should be able to configure the mounting of a filesystem.

Key Knowledge Areas:

- Manually mount and unmount filesystems.
- Configure filesystem mounting on bootup.
- Configure user mountable removable filesystems.
- Use of labels and UUIDs for identifying and mounting file systems.
- Awareness of systemd mount units.

The following is a partial list of the used files, terms and utilities:

- /etc/fstab
- /media/
- mount
- umount
- blkid
- lsblk

104.4 Removed

104.5 Manage file permissions and ownership

Weight: 3

Description: Candidates should be able to control file access through the proper use of permissions and ownerships.

Key Knowledge Areas:

- Manage access permissions on regular and special files as well as directories.
- Use access modes such as suid, sgid and the sticky bit to maintain security.
- Know how to change the file creation mask.
- Use the group field to grant file access to group members.

The following is a partial list of the used files, terms and utilities:

- chmod
- umask
- chown
- chgrp

104.6 Create and change hard and symbolic links

Weight: 2

Description: Candidates should be able to create and manage hard and symbolic links to a file.

Key Knowledge Areas:

- Create links.
- Identify hard and/or soft links.
- Copying versus linking files.
- Use links to support system administration tasks.

The following is a partial list of the used files, terms and utilities:

- ln
- ls

104.7 Find system files and place files in the correct location

Weight: 2

Description: Candidates should be thoroughly familiar with the Filesystem Hierarchy Standard (FHS), including typical file locations and directory classifications.

Key Knowledge Areas:

- Understand the correct locations of files under the FHS.
- Find files and commands on a Linux system.
- Know the location and purpose of important file and directories as defined in the FHS.

The following is a partial list of the used files, terms and utilities:

- find
- locate
- updatedb
- whereis
- which
- type
- /etc/updatedb.conf

Topic 105: Shells and Shell Scripting

105.1 Customize and use the shell environment

Weight: 4

Description: Candidates should be able to customize shell environments to meet users' needs. Candidates should be able to modify global and user profiles.

Key Knowledge Areas:

- Set environment variables (e.g. PATH) at login or when spawning a new shell.
- Write Bash functions for frequently used sequences of commands.
- Maintain skeleton directories for new user accounts.
- Set command search path with the proper directory.

The following is a partial list of the used files, terms and utilities:

- .
- source
- /etc/bash.bashrc
- /etc/profile
- env
- export
- set
- unset
- ~/.bash_profile
- ~/.bash_login
- ~/.profile
- ~/.bashrc
- ~/.bash_logout
- function
- alias

105.2 Customize or write simple scripts

Weight: 4

Description: Candidates should be able to customize existing scripts, or write simple new Bash scripts.

Key Knowledge Areas:

- Use standard sh syntax (loops, tests).
- Use command substitution.
- Test return values for success or failure or other information provided by a command.
- Execute chained commands.
- Perform conditional mailing to the superuser.
- Correctly select the script interpreter through the shebang (#!) line.
- Manage the location, ownership, execution and suid-rights of scripts.

The following is a partial list of the used files, terms and utilities:

- for
- while
- test
- if
- read
- seq
- exec
- ||
- &&

Topic 106: User Interfaces and Desktops

106.1 Install and configure X11

Weight: 2

Description: Candidates should be able to install and configure X11.

Key Knowledge Areas:

- Understanding of the X11 architecture.
- Basic understanding and knowledge of the X Window configuration file.
- Overwrite specific aspects of Xorg configuration, such as keyboard layout.
- Understand the components of desktop environments, such as display managers and window managers.
- Manage access to the X server and display applications on remote X servers.
- Awareness of Wayland.

The following is a partial list of the used files, terms and utilities:

- /etc/X11/xorg.conf
- /etc/X11/xorg.conf.d/
- ~/.xsession-errors
- xhost
- xauth
- DISPLAY
- X

106.2 Graphical Desktops

Weight: 1

Description: Candidates should be aware of major Linux desktops. Furthermore, candidates should be aware of protocols used to access remote desktop sessions.

Key Knowledge Areas:

- Awareness of major desktop environments
- Awareness of protocols to access remote desktop sessions

The following is a partial list of the used files, terms and utilities:

- KDE
- Gnome
- Xfce
- X11
- XDMCP
- VNC
- Spice
- RDP

106.3 Accessibility

Weight: 1

Description: Demonstrate knowledge and awareness of accessibility technologies.

Key Knowledge Areas:

- Basic knowledge of visual settings and themes.
- Basic knowledge of assistive technology.

The following is a partial list of the used files, terms and utilities:

- High Contrast/Large Print Desktop Themes.
- Screen Reader.
- Braille Display.
- Screen Magnifier.
- On-Screen Keyboard.
- Sticky/Repeat keys.
- Slow/Bounce/Toggle keys.
- Mouse keys.
- Gestures.
- Voice recognition.

Topic 107: Administrative Tasks

107.1 Manage user and group accounts and related system files

Weight: 5

Description: Candidates should be able to add, remove, suspend and change user accounts.

Key Knowledge Areas:

- Add, modify and remove users and groups.
- Manage user/group info in password/group databases.
- Create and manage special purpose and limited accounts.

The following is a partial list of the used files, terms and utilities:

- /etc/passwd
- /etc/shadow
- /etc/group
- /etc/skel/
- chage
- getent
- groupadd
- groupdel
- groupmod
- passwd
- useradd
- userdel
- usermod

107.2 Automate system administration tasks by scheduling jobs

Weight: 4

Description: Candidates should be able to use cron and systemd timers to run jobs at regular intervals and to use at to run jobs at a specific time.

Key Knowledge Areas:

- Manage cron and at jobs.
- Configure user access to cron and at services.
- Understand systemd timer units.

The following is a partial list of the used files, terms and utilities:

- /etc/cron.{d,daily,hourly,monthly,weekly}/
- /etc/at.deny
- /etc/at.allow
- /etc/crontab
- /etc/cron.allow
- /etc/cron.deny
- /var/spool/cron/
- crontab
- at
- atq
- atrm
- systemctl
- systemd-run

107.3 Localisation and internationalisation

Weight: 3

Description: Candidates should be able to localize a system in a different language than English. As well, an understanding of why LANG=C is useful when scripting.

Key Knowledge Areas:

- Configure locale settings and environment variables.
- Configure timezone settings and environment variables.

The following is a partial list of the used files, terms and utilities:

- /etc/timezone
- /etc/localtime
- /usr/share/zoneinfo/
- LC_*
- LC_ALL
- LANG
- TZ
- /usr/bin/locale
- tzselect
- timedatectl
- date
- iconv
- UTF-8
- ISO-8859
- ASCII
- Unicode

Topic 108: Essential System Services

108.1 Maintain system time

Weight: 3

Description: Candidates should be able to properly maintain the system time and synchronize the clock via NTP.

Key Knowledge Areas:

- Set the system date and time.
- Set the hardware clock to the correct time in UTC.
- Configure the correct timezone.
- Basic NTP configuration using ntpd and chrony.
- Knowledge of using the pool.ntp.org service.
- Awareness of the ntpq command.

The following is a partial list of the used files, terms and utilities:

- /usr/share/zoneinfo/
- /etc/timezone
- /etc/localtime
- /etc/ntp.conf
- /etc/chrony.conf
- date
- hwclock
- timedatectl
- ntpd
- ntpdate
- chronyc
- pool.ntp.org

108.2 System logging

Weight: 4

Description: Candidates should be able to configure rsyslog. This objective also includes configuring the logging daemon to send log output to a central log server or accept log output as a central log server. Use of the systemd journal subsystem is covered. Also, awareness of syslog and syslog-ng as alternative logging systems is included.

Key Knowledge Areas:

- Basic configuration of rsyslog.
- Understanding of standard facilities, priorities and actions.
- Query the systemd journal.
- Filter systemd journal data by criteria such as date, service or priority.
- Configure persistent systemd journal storage and journal size.
- Delete old systemd journal data.
- Retrieve systemd journal data from a rescue system or file system copy.
- Understand interaction of rsyslog with systemd-journald.
- Configuration of logrotate.
- Awareness of syslog and syslog-ng.

Terms and Utilities:

- /etc/rsyslog.conf
- /var/log/
- logger
- logrotate
- /etc/logrotate.conf
- /etc/logrotate.d/
- journalctl

- systemd-cat
- /etc/systemd/journald.conf
- /var/log/journal/

108.3 Mail Transfer Agent (MTA) basics

Weight: 3

Description: Candidates should be aware of the commonly available MTA programs and be able to perform basic forward and alias configuration on a client host. Other configuration files are not covered.

Key Knowledge Areas:

- Create e-mail aliases.
- Configure e-mail forwarding.
- Knowledge of commonly available MTA programs (postfix, sendmail, exim) (no configuration).

Terms and Utilities:

- ~/.forward
- sendmail emulation layer commands
- newaliases
- mail
- mailq
- postfix
- sendmail
- exim

108.4 Manage printers and printing

Weight: 2

Description: Candidates should be able to manage print queues and user print jobs using CUPS and the LPD compatibility interface.

Key Knowledge Areas:

- Basic CUPS configuration (for local and remote printers).
- Manage user print queues.
- Troubleshoot general printing problems.
- Add and remove jobs from configured printer queues.

The following is a partial list of the used files, terms and utilities:

- CUPS configuration files, tools and utilities
- /etc/cups/
- lpd legacy interface (lpr, lprm, lpq)

Topic 109: Networking Fundamentals

109.1 Fundamentals of internet protocols

Weight: 4

Description: Candidates should demonstrate a proper understanding of TCP/IP network fundamentals.

Key Knowledge Areas:

- Demonstrate an understanding of network masks and CIDR notation.
- Knowledge of the differences between private and public “dotted quad” IP addresses.
- Knowledge about common TCP and UDP ports and services (20, 21, 22, 23, 25, 53, 80, 110, 123, 139, 143, 161, 162, 389, 443, 465, 514, 636, 993, 995).
- Knowledge about the differences and major features of UDP, TCP and ICMP.
- Knowledge of the major differences between IPv4 and IPv6.
- Knowledge of the basic features of IPv6.

The following is a partial list of the used files, terms and utilities:

- /etc/services
- IPv4, IPv6
- Subnetting
- TCP, UDP, ICMP

109.2 Persistent network configuration

Weight: 4

Description: Candidates should be able to manage the persistent network configuration of a Linux host.

Key Knowledge Areas:

- Understand basic TCP/IP host configuration.
- Configure ethernet and wi-fi network using NetworkManager.
- Awareness of systemd-networkd.

The following is a partial list of the used files, terms and utilities:

- /etc/hostname
- /etc/hosts
- /etc/nsswitch.conf
- /etc/resolv.conf
- nmcli
- hostnamectl
- ifup
- ifdown

109.3 Basic network troubleshooting

Weight: 4

Description: Candidates should be able to troubleshoot networking issues on client hosts.

Key Knowledge Areas:

- Manually configure network interfaces, including viewing and changing the configuration of network interfaces using iproute2.
- Manually configure routing, including viewing and changing routing tables and setting the default route using iproute2.
- Debug problems associated with the network configuration.
- Awareness of legacy net-tools commands.

The following is a partial list of the used files, terms and utilities:

- ip
- hostname
- ss
- ping
- ping6
- traceroute
- traceroute6
- tracepath
- tracepath6
- netcat
- ifconfig
- netstat
- route

109.4 Configure client side DNS

Weight: 2

Description: Candidates should be able to configure DNS on a client host.

Key Knowledge Areas:

- Query remote DNS servers.
- Configure local name resolution and use remote DNS servers.
- Modify the order in which name resolution is done.
- Debug errors related to name resolution.
- Awareness of systemd-resolved.

The following is a partial list of the used files, terms and utilities:

- /etc/hosts
- /etc/resolv.conf
- /etc/nsswitch.conf
- host
- dig
- getent

Topic 110: Security

110.1 Perform security administration tasks

Weight: 3

Description: Candidates should know how to review system configuration to ensure host security in accordance with local security policies.

Key Knowledge Areas:

- Audit a system to find files with the suid/sgid bit set.
- Set or change user passwords and password aging information.
- Being able to use nmap and netstat to discover open ports on a system.
- Set up limits on user logins, processes and memory usage.
- Determine which users have logged in to the system or are currently logged in.
- Basic sudo configuration and usage.

The following is a partial list of the used files, terms and utilities:

- find
- passwd
- fuser
- lsof
- nmap
- chage
- netstat
- sudo
- /etc/sudoers
- su
- usermod
- ulimit
- who, w, last

110.2 Setup host security

Weight: 3

Description: Candidates should know how to set up a basic level of host security.

Key Knowledge Areas:

- Awareness of shadow passwords and how they work.
- Turn off network services not in use.
- Understand the role of TCP wrappers.

The following is a partial list of the used files, terms and utilities:

- /etc/nologin
- /etc/passwd
- /etc/shadow
- /etc/xinetd.d/
- /etc/xinetd.conf
- systemd.socket
- /etc/inittab
- /etc/init.d/
- /etc/hosts.allow
- /etc/hosts.deny

110.3 Securing data with encryption

Weight: 4

Description: The candidate should be able to use public key techniques to secure data and communication.

Key Knowledge Areas:

- Perform basic OpenSSH 2 client configuration and usage.
- Understand the role of OpenSSH 2 server host keys.
- Perform basic GnuPG configuration, usage and revocation.
- Use GPG to encrypt, decrypt, sign and verify files.
- Understand SSH port tunnels (including X11 tunnels).

The following is a partial list of the used files, terms and utilities:

- ssh
- ssh-keygen
- ssh-agent
- ssh-add
- ~/.ssh/id_rsa and id_rsa.pub
- ~/.ssh/id_dsa and id_dsa.pub
- ~/.ssh/id_ecdsa and id_ecdsa.pub
- ~/.ssh/id_ed25519 and id_ed25519.pub
- /etc/ssh/ssh_host_rsa_key and ssh_host_rsa_key.pub
- /etc/ssh/ssh_host_dsa_key and ssh_host_dsa_key.pub
- /etc/ssh/ssh_host_ecdsa_key and ssh_host_ecdsa_key.pub
- /etc/ssh/ssh_host_ed25519_key and ssh_host_ed25519_key.pub
- ~/.ssh/authorized_keys
- ssh_known_hosts
- gpg
- gpg-agent
- ~/.gnupg/

Future Change Considerations

Future changes to the objective will/may include:

- Remove ifup/ifdown and legacy net-tools command
- Remove TCP wrappers

سرفصل‌های دوره آموزشی

Embedded Linux

(Level-1)

Introduction to embedded Linux

- Advantages of Linux versus traditional embedded operating systems
- Typical hardware platforms used to run embedded Linux systems
- Overall architecture of embedded Linux systems
- Overview of the major software components
- Development environment for Embedded Linux development

Cross-compiling toolchain and C library

- What is inside a cross-compiling toolchain?
- Choosing the target C library
- What is inside the C LIBRARY?
- Ready to use cross-compiling toolchains
- Building a cross-compiling toolchain with automated tools
- Getting and configuring Crosstool-NG
- Executing it to build a custom cross compilation toolchain
- Exploring the contents of the toolchain

Boot process, firmware, and bootloaders

- Booting process of embedded platforms, focus on the x86 and ARM architectures
- Boot process and bootloaders on x86 platforms (legacy and UEFI)
- Boot process on ARM platforms: ROM code, bootloaders, ARM Trusted Firmware
- Focus on U-Boot: configuration, installation, and usage.
- U-Boot commands, U-Boot environment, U-Boot scripts, U-Boot generic distro boot mechanism

Bootloader and U-boot

- Set up serial communication with the board.
- Configure, compile and install U-Boot for the target hardware
- Configure, compile and install Trusted Firmware-A
- Become familiar with U-Boot environment and commands
- Set up TFTP communication with the board. Use TFTP U-Boot commands

Linux kernel

- Role and general architecture of the Linux kernel
- Separation between kernel and user-space, and interfaces between user-space and the Linux kernel
- Understanding Linux kernel versions choosing between vendor-provided kernel and upstream kernel, Long Term Support versions
- Getting the Linux kernel source code
- Fetching Linux kernel sources
- Clone the mainline Linux tree
- Accessing stable releases

Configuring, compiling and booting the Linux kernel

- Configuring the Linux kernel: ready-made configuration files, configuration interfaces
- Concept of Device Tree
- Cross-compiling the Linux kernel
- Study of the generated files and their role
- Installing and booting the Linux kernel
- The Linux kernel command line
- Configuring the Linux kernel and cross-compiling it for the embedded hardware
- Downloading your kernel on the board through U-boot’s TFTP client.
- Booting your kernel.
- Automating the kernel boot process with UBoot

Root filesystem in Linux

- Filesystems in Linux
- Role and organization of the root filesystem
- Location of the root filesystem: on storage, in memory, from the network
- Device files, virtual filesystems
- Contents of a typical root filesystem
- Detailed overview. Detailed features
- Configuration, compiling and deploying

Tiny root file system built from scratch with BusyBox

- Setting up a kernel to boot your system on a workstation directory exported by NFS
- Passing kernel command line parameters to boot on NFS
- Creating the full root filesystem from scratch. Populating it with BusyBox based utilities.
- System startup using BusyBox init
- Using the BusyBox HTTP server.
- Controlling the target from a web browser on the PC host.

Block filesystems

- Accessing and partitioning block devices
- Filesystems for block devices
- Usefulness of journaled filesystems
- Read-only block filesystems
- RAM filesystems
- How to create each of these filesystems
- Suggestions for embedded systems
- Creating partitions on your SD card
- Booting a system with a mix of filesystems: SquashFS for the root filesystem, ext4 for system data, and tmpfs for temporary system files

Flash filesystems

- The Memory Technology Devices (MTD) filesystem
- Filesystems for MTD storage: JFFS2, Yaffs2, UBIFS
- Kernel configuration options
- MTD storage partitions
- Focus on today’s best solution, UBI and UBIFS: preparing, flashing and using UBI images

سرفصل‌های دوره آموزشی

Embedded Linux

(Level-2)

Root filesystem in Linux

- Filesystems in Linux
- Role and organization of the root filesystem
- Location of the root filesystem: on storage, in memory & from the network
- Device files, virtual filesystems
- Contents of a typical root filesystem
- Overview & features
- Configuration, compiling and deploying

Tiny root filesystem built from scratch with BusyBox

- Setting up a kernel to boot your system on a workstation directory exported by NFS
- Passing kernel command line parameters to boot on NFS
- Creating the full root filesystem from scratch. Populating it with BusyBox based utilities.
- System startup using BusyBox init
- Using the BusyBox HTTP server

- Controlling the target from a web browser on the PC host
- Setting up shared libraries on the target and compiling a sample executable

Accessing hardware devices

- How to access hardware on popular busses: USB, SPI, I2C, PCI
- Usage of kernel drivers and direct userspace access
- The Device Tree syntax, and how to use it to describe additional devices
- Finding Linux kernel drivers for specific hardware devices
- Using kernel modules
- Hardware access using /dev and /sys
- User-space interfaces for the most common hardware devices: storage, network, GPIO, LEDs, audio, graphics, video
- Exploring the contents of /dev and /sys and the devices available on the embedded hardware platform
- Using GPIOs and LEDs
- Modifying the Device Tree to control pin multiplexing and to declare an I2Cconnected joystick
- Adding support for a USB audio card using
- Linux kernel modules
- Adding support for the I2C-connected joystick through an out-of-tree module

Cross-compiling user-space libraries and applications

- Configuring, cross-compiling and installing applications and libraries
- Concept of build system, and overview of a few common build systems used by open-source projects: Makefile, autotools, CMake, meson
- Overview of the common issues encountered when cross-compiling Manual cross-compilation of several opensource libraries and applications for an embedded platform

- Learning about common pitfalls and issues, and their solutions
- This includes compiling alsa-utils package, and using its speaker-test program to test that audio works on the target

Embedded system building tools

- Approaches for building embedded Linux systems: build systems and binary distributions
- Principle of build systems, overview of Yocto Project/OpenEmbedded and Buildroot.
- Principle of binary distributions and useful tools, focus on Debian/ Ubuntu
- Specialized software frameworks/ distributions: Tizen, AGL, Android
- Using Buildroot to rebuild the same basic system plus a sound playing server (MPD) and a client to control it (mpc)
- Driving music playback, directly from the target, and then remotely through an MPD client on the host machine
- Analyzing dependencies between packages

Open source licenses and compliance

- Presentation of the most important open-source licenses: GPL, LGPL, MIT, BSD, Apache and etc.
- Concept of copyleft licenses
- Differences between (L) GPL version 2 and 3
- Compliance with open-source licenses: best practices

Overview of major embedded Linux software stacks

- Systemd as an init system
- Hardware management with udev
- Inter-process communication with D-Bus
- The connectivity software stack: Ethernet, WiFi, modems, Bluetooth
- The graphics software stack: DRM/KMS, X.org, Wayland, Qt, Gtk, OpenGL
- The multimedia software stack: Video4Linux, GStreamer, Pulseaudio, Pipewire

Integration of additional software stacks

- Integration of systemd as an init system
- Use udev built in systemd for automatic module loading

Application development and debugging

- Programming languages and libraries available.
- Build system for your application, an overview of CMake and meson
- The gdb debugger: remote debugging with gdbserver, post-mortem debugging with core files
- Performance analysis, tracing and profiling tools, memory checkers: strace, ltrace, perf, valgrind
- Creating an application that uses an I2Cconnected joystick to control an audio player
- Setting up an IDE to develop and remotely debug an application
- Using strace, ltrace, gdbserver and perf to debug/investigate buggy applications on the embedded board

Useful resources

- Books about embedded Linux and system programming
- Useful online resources
- International conferences