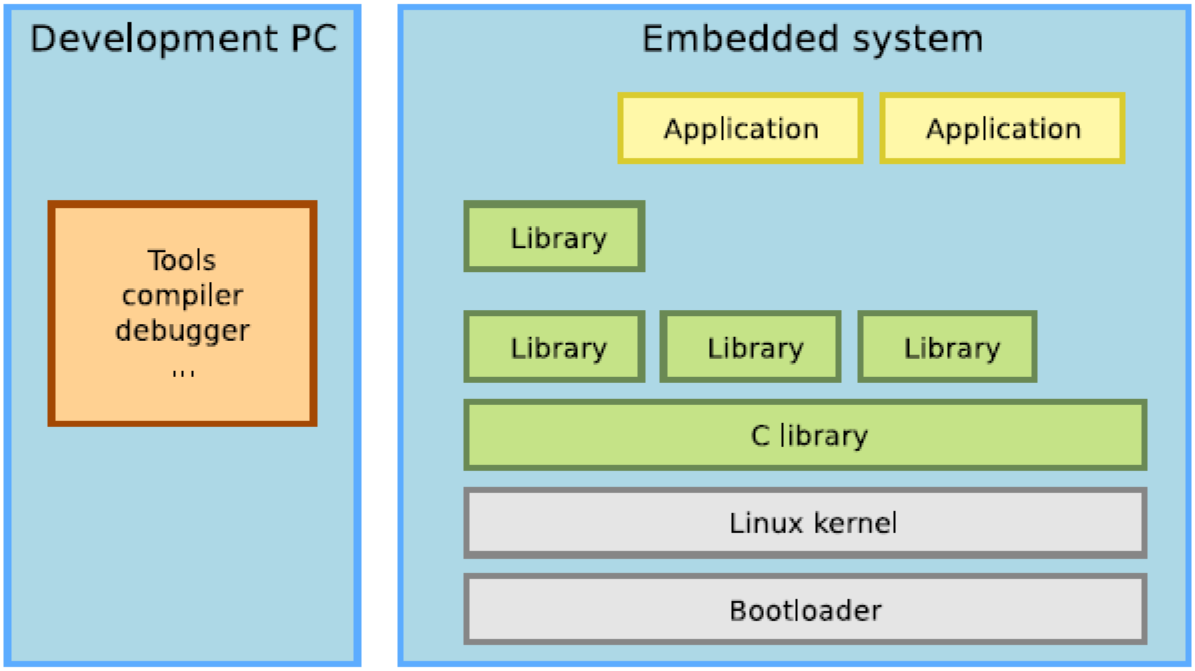
1/ INVESTIGATE LINUX

1.1/ Overview about Embedded Linux.

- Embedded Linux is the usage of the Linux kernel and various open-source components in embedded systems.
- Advantages of Linux and open-source for embedded systems: re-using components (allows to focus on the added value of your products), low cost (allows to have a higher budget for the hardware or to increase the company's skills and knowledge), full control (allows to have full control over the software part of your system), quality (allows to design your system with high quality components at the foundations), eases testing at new features (allows to easily explore new possibilities and solutions), community support and taking part into the community.
- The Linux kernel and most other architecture-dependent components support a wide range of 32 and 64 bit architectures.
- Linux is not designed for small microcontrollers. The memory is 8MB but usually require at least 32 MB of RAM, the minimum of storage is within 4MB but more is needed.
- Type of hardware platforms:
  + Evaluation platforms form SoC vendor.
  + Component on Module.
  + Community development platforms.
  + Custom platforms.
- Criteria for choosing the hardware: between properly supported hardware in the official Linux kernel and poorly-supported hardware, there will be huge differences in development time and cost.

- Embedded Linux system architecture:



- The picture above is included:

+ Cross-compilation toolchain: compiler that runs on the development machine, but generates code for the target.

+ Bootloader: started by the hardware, responsible for basic initialization, loading and executing the kernel.

+ Linux Kernel: contains the process and memory management, network stack, devices drivers and provides services to user space applications.

+ C library: the interface between the kernel and the user space

+ Applications.: libraries and applications

+ Third-party or in-house

1.2/ The Linux and GNU/Linux command line

- Almost everything in Unix is a file (Regular files, Directories (files listing a set of files), Symbolic links (files referring to the name of another file), Devices and Peripherals (read and write form devices as with regular files), Pipes, Sockets (inter process communications)).
- File name features since the beginning of Unix: you can do anything and any contain character except “/”.

- File path is a sequence of nested directories with a file or directories or directories at the end, and separated by “/” character.
- Shell is a tool to execute user commands. Now bash (The Bourne Again shell) is the most popular used in the world.

2/ BASIC TOOL/COMMAND IN LINUX

- Summary of most useful commands.

2.1/ Handing a files and directories.

Create a directory:

```
mkdir unixstuff
```

Create nested directories:

```
mkdir -p dir1/dir2
```

Changing directories:

```
Cd unixstuff
cd .. (parent directory)
cd - (previous directory)
cd (home directory)
cd ~bill (home directory of user bill)
```

Print the working (current) directory:

```
pwd
```

Copy a file to another:

```
cp /home/mobaxterm/unistuff/dangle.txt .
```

Copy files to a directory:

```
cp dangle.txt dangle.bak
```

Copy directories recursively:

```
cp -r unixstuff backups
rsync -a source_dir/ dest_dir/
```

Create a symbolic link:

```
ln -s dangle.txt link
```

Rename a file, link or directory:

```
mv dangle.txt dangle1.txt
```

Remove files or links:

```
rm file1 file2
```

Remove empty directories:

```
rmdir dir
```

Remove non-empty directories:

```
rm -rf backups
```

2.2/ Listing files

List all “regular” files (not starting with) in the current directory:

```
ls
```

Display a long listing:

```
ls -l
```

List all the files in the current directory, including “hidden” ones (starting with):

```
ls -a
```

List by time (most recent files first):

```
ls -t
```

List by size (biggest files first)

```
ls -S
```

List with a reverse sort order:

```
ls –r
```

Long list with most recent files last:

```
ls –ltr
```

2.3/ Displaying file contents

Concatenate and display file contents:

```
cat file1 file2
```

Display the contents of several files (stopping at each page):

```
more file1 file2
```

`less file1 file2` (better: extra features)

Display the first 10 lines of a file:

```
head -10 file
```

Display the last 10 lines of a file:

```
tail -10 file
```

2.4/ File name pattern matching

Concatenate all “regular” files:

```
cat  *
```

Concatenate all “hidden” files:

```
cat  *
```

Concatenate all files ending with .log:

```
cat *.log
```

List “regular” files with bug in their name:

```
ls *bug*
```

List all “regular” files ending with. and a single character:

```
ls *.?
```

2.5/ handing file contents

Show only the lines in a file containing a given substring:

```
grep substring file
```

Case insensitive search:

```
grep -i substring file
```

Showing all the lines but the ones containing a substring:

```
grep -v substring file
```

Search through all the files in a directory:

```
grep -r substring dir
```

Sort lines in a given file:

```
sort file
```

Sort lines, only display duplicate ones once:

```
sort -u file
```

(unique)

2.6/ Changing file access rights.

Add write permissions to the current user:

```
chmod u+w file
```

Add read permissions to users in the file group:

```
chmod g+r file
```

Add execute permissions to other users:

```
chmod o+x file
```

Add read + write permissions to all users:

```
chmod a+rw file
```

Make executable files executable by all:

```
chmod a+rX *
```

Make the whole directory and its contents accessible by all users:

`chmod -R a+rX dir` (**r**ecursive)

2.7 Comparing files and directories

Comparing 2 files:

```
diff file1 file2
```

Comparing 2 files (graphical):

```
gvimdiff file1 file2
tkdiff file1 file2
meld file1 file2
```

Comparing 2 directories:

```
diff -r dir1 dir2
```

2.8/ Looking for files.

Find all files in the current (.) directory and its subdirectories with `log` in their name:

```
find . -name “*log*”
```

Find all the `.pdf` files in `dir` and subdirectories and run a command on each:

```
find . -name “*.pdf” -exec xpdf {} ';'
```

Quick system-wide file search by pattern (caution: index based, misses new files):

```
locate “*pub*”
```

2.9/ Redirecting command output

Redirect command output to a file:

```
ls *.png > image_files
```

Append command output to an existing file:

```
ls *.jpg >> image_files
```

Redirect command output to the input of another command:

```
cat *.log | grep error
```

2.10/ Job control

Show all running processes:

```
ps -ef
```

Live hit-parade of processes (press `P`, `M`, `T`: sort by Processor, Memory or Time usage):

```
top
```

Send a termination signal to a process:

`kill <pid>` (number found in `ps` output)

Have the kernel kill a process:

```
kill -9 <pid>
```

Kill all processes (at least all user ones):

```
kill -9 -1
```

Kill a graphical application:

`xkill` (click on the program window to kill)

2.11/File and partition sizes

Show the total size on disk of files or directories (**d**isk **u**sage):

```
du -sh dir1 dir2 file1 file2
```

Number of bytes, words and lines in file:

`wc file` (**w**ord **c**ount)

Show the size, total space and free space of the current partition:

```
df -h .
```

Display these info for all partitions:

```
df –h
```

2.12/ Compressing

Compress a file:

`gzip file` (`.gz` format)

`bzip2 file` (`.bz2` format, better)

`lzma file` (`.lzma` format, best compression)

`xz file` (`.xz` format, best for code)

Uncompress a file:

```
gunzip file.gz
bunzip2 file.bz2
unlzma file.lzma
unxz file.xz
```

2.13/ Archiving

Create a compressed archive (**ta**pe **ar**chive):

```
tar zcvf archive.tar.gz dir
```

```
tar jcvf archive.tar.bz2 dir
```

```
tar Jcvf archive.tar.xz dir
```

```
tar --lzma -cvf archive.tar.lzma
```

Test (list) a compressed archive:

```
tar tvf archive.tar.[gz|bz2|lzma|xz]
```

E**x**tract the contents of a compressed archive:

```
tar xvf archive.tar.[gz|bz2|lzma|xz]
```

tar options:

c: **c**reate

t: **t**est

x: e**x**tract

j: on the fly bzip2 (un)compression

J: on the fly xz (un)compression

z: on the fly gzip (un)compression

Handling zip archives

```
zip -r archive.zip <files>
```
(create)

```
unzip -t archive.zip
```
(**t**est / list)

```
unzip archive.zip
```
(extract)

2.14/ Printing

Send PostScript or text files to queue:

```
lpr -Pqueue f1.ps f2.txt
```
(**l**ocal **pr**inter)

List all the print jobs in queue:

```
lpq -Pqueue
```

Cancel a print job number in queue:

```
cancel 123 queue
```

Print a PDF file:

```
pdf2ps doc.pdf
lpr doc.ps
```

View a PostScript file:

```
ps2pdf doc.ps
```

```
xpdf doc.pdf
```

2.15/ User management

List users logged on the system:

```
who
```

Show which user I am logged as:

```
whoami
```

Show which groups user belongs to:

```
groups user
```

Tell more information about user:

```
finger user
```

Switch to user hulk:

```
su - hulk
```

Switch to super user (root):

`su -` (**s**witch **u**ser)

`su` (keep same directory and environment)

2.16/ Time management

Wait for 60 seconds:

```
sleep 60
```

Show the current date:

```
date
```

Count the time taken by a command:

```
time find_charming_prince -cute -rich
```

2.17 Command help

Basic help (works for most commands):

```
grep –help
```

Access the full manual page of a command:

```
man grep
```

2.18/ Misc commands

Basic command-line calculator

```
bc -l
```

2.19/ Basic system administration

Change the owner and group of a directory and all its contents:

```
sudo chown -R newuser.newgroup dir
```

Reboot the machine in 5 minutes:

```
sudo shutdown -r +5
```

Shutdown the machine now:

```
sudo shutdown -h now
```

Display all available network interfaces:

```
ifconfig -a
```

Assign an IP address to a network interface:

```
sudo ifconfig eth0 207.46.130.108
```

Bring down a network interface:

```
sudo ifconfig eth0 down
```

Define a default gateway for packets to machines outside the local network:

```
sudo route add default gw 192.168.0.1
```

Delete the default route:

```
sudo route del default
```

Test networking with another machine:

```
ping 207.46.130.108
```

Create or remove partitions on the first IDE hard disk:

```
fdisk /dev/hda1
```

Create (format) an ext3 filesystem:

```
mkfs.ext3 /dev/hda1
```

Create (format) a FAT32 filesystem:

```
mkfs.vfat -v -F 32 /dev/hda2
```

Mount a formatted partition:

```
mkdir /mnt/usbdisk (just do it once)
sudo mount /dev/uba1 /mnt/usbdisk
```

Mount a filesystem image (loop device):

```
sudo mount -o loop fs.img /mnt/fs
```

Unmount a filesystem:

```
sudo umount /mnt/usbdisk
```

Check the system kernel version:

```
uname -a
```