

P5 리눅스 시스템 빌드 파일 시스템 빌드

소프트웨어융합 김태민

1. 파일 시스템 개요

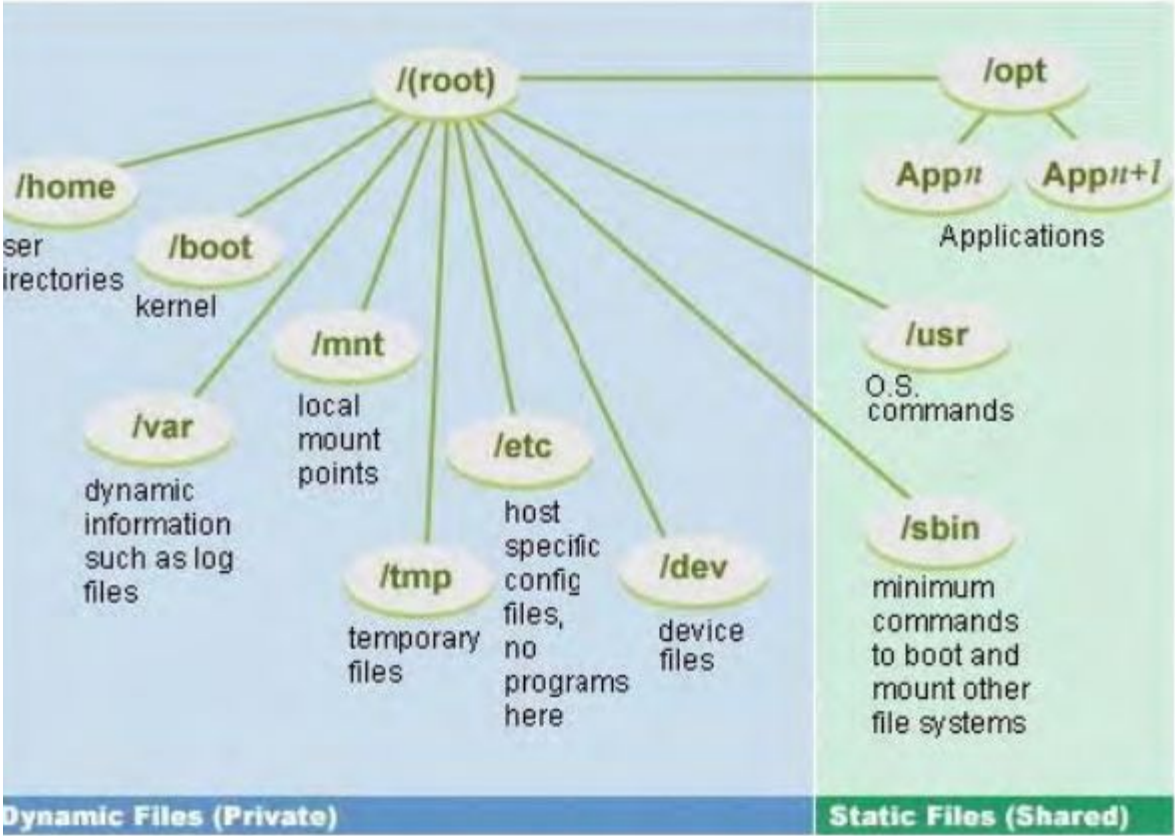
- 1) 임베디드 리눅스 파일 시스템의 사용 목적과 기능에 대해서 설명하시오.
- 2) 파일 시스템의 디렉토리 구조에 대해 설명하시오.

1) 임베디드 리눅스 파일시스템의 사용 목적과 기능에 대해서 설명하시오.

- 파일 시스템은 리눅스와 같은 운영체제가 파일을 저장하기 위해 파일을 관리하는 체계를 제공한다. 파일 시스템의 목적은 하드디스크와 메인 메모리의 속도 차를 줄이고, 파일을 관리하고 하드디스크의 용량을 효율적으로 이용하는 것이다. 운영체제에 따라 호환되지 않는 자신만의 파일시스템을 만들어 운영하게 된다.
- 파일 시스템의 기능은 사용자가 파일을 생성/삭제/수정할 수 있도록 하고, 여러 사용자가 파일을 공유할 수 있도록 한다. 또한 파일에 대한 백업과 복구를 할 수 있도록 도와준다. 뿐만 아니라 주기억장치와 보조기억장치 간의 파일 전송을 담당하고 있다.

2) 파일 시스템의 디렉토리 구조에 대해 설명하시오

리눅스의 기본 디렉토리 구조는 아래의 그림과 같이 트리 구조를 하고 있으며, 기본 구조를 제외한 나머지는 사용자가 설정하여 변경이 가능하다. 리눅스의 디렉토리 구조는 파일 시스템 표준안을 기반으로 구성되며 대부분 리눅스는 **FHS** 표준 파일 시스템 계층을 사용하고 같은 목적으로 쓰이는 파일들은 같은 장소에 모아서 관리한다. 이는 시스템의 자원이나 프로그램을 쉽게 찾을 수 있도록 도와준다.



2) 파일 시스템의 디렉토리 구조에 대해 설명하시오

주요 디렉토리의 특징

`/`: 루트 디렉토리로 리눅스 시스템에서 가장 최상위 디렉토리

`/bin`: 리눅스에서 가장 기본이 되는 명령어들이 모여있는 이진파일로 이루어진 디렉토리

`/etc`: 리눅스 시스템에 관한 각종 환경 설정에 연관된 파일들과 디렉토리들을 가짐

`/etc/shadow`: 파일에서 패스워드 부분만을 따로 저장한 디렉토리

`/etc/group`: 시스템의 그룹에 대한 정보를 저장하고 있는 파일

`/etc/inittab`: `init`를 설정하는 파일

`etc/issue`: `getty`에 의해 로그인을 위한 프롬프트가 뜨기 전에 출력되는 메시지를 설정하는 파일

`etc/profile`: 시스템이 시작될 때 사용자가 로그인을 할 때 본셸이나 C셸에 의해 실행되는 스크립트 파일

`etc/securetty`: 시스템 관리자가 시스템에 로그인할 수 있는 안전한 터미널에 대한 정보가 저장되어 있는 파일

`/mnt`: 외부 장치인 플로피 디스크, 시디롬 등을 마운트하기 위해 제공되는 디렉토리

`/usr`: 시스템에 사용되는 각종 프로그램들이 설치되는 디렉토리

`/usr/bin`: 리눅스 시스템에서 사용되는 각종 프로그램들이 저장되어 있는 디렉토리

2) 파일 시스템의 디렉토리 구조에 대해 설명하시오

/usr/sbin: 시스템 관리자를 위한 명령어들이 저장되는 디렉토리

usr/lib: 각종 라이브러리들이 저장되어 있는 디렉토리

/lib: 프로그램들의 각종 라이브러리가 존재하는 디렉토리

/home: 시스템 계정 사용자들의 홈 디렉토리 및 **ftp**와 같은 서비스 디렉토리들이 저장된 곳

/dev: 시스템의 각종 디바이스들에 접근하기 위한 디바이스 드라이버들이 저장된 디렉토리

/dev/console: 시스템의 콘솔

/dev/hda: 시스템의 하드 디스크

/dev/lp: 시스템의 병렬 포트 장치

/dev/null: 데이터 등을 보내면 폐기시키는 디렉토리

/dev/pty: 시스템으로의 원격 접속을 위한 **pseudo-terminal**

/dev/sda: SCSI 장치들

2) 파일 시스템의 디렉토리 구조에 대해 설명하시오

/dev/ttyS: 직렬포트 장치

/dev/cuaS: Callout. 장치

/dev/tty: 시스템의 가상콘솔

var: 시스템에서 사용되는 동적인 파일들이 저장되는 디렉토리

/var/cache: 포맷된 매뉴얼 페이지들이 잠시 대기하기 위한 디렉토리

/var/lib: 시스템이 동작하면서 계속 수정되고 변경하는 파일들을 위한 디렉토리

/var/local: **/usr/local** 디렉토리에 설치된 프로그램들의 각종 데이터들이 저장되는 디렉토리

/var/lock: 잠금 파일들이 저장되는 디렉토리

/var/log: 프로그램들의 로그 파일들이 저장되는 디렉토리

/var/run: 시스템의 현재 정보들을 저장하고 있는 디렉토리

/var/spool: 메일이나 뉴스, 프린터 큐 등과 같은 시스템 상에서 대기 상태에 있는 작업들을 위한 디렉토리

/var/tmp: **/tmp**에 저장된 임시 파일들 중에 오래 보관되어야 할 임시 파일들이 저장되는 디렉토리

/tmp: 임시 파일들을 위한 디렉토리

/root: 시스템 관리자의 홈 디렉토리

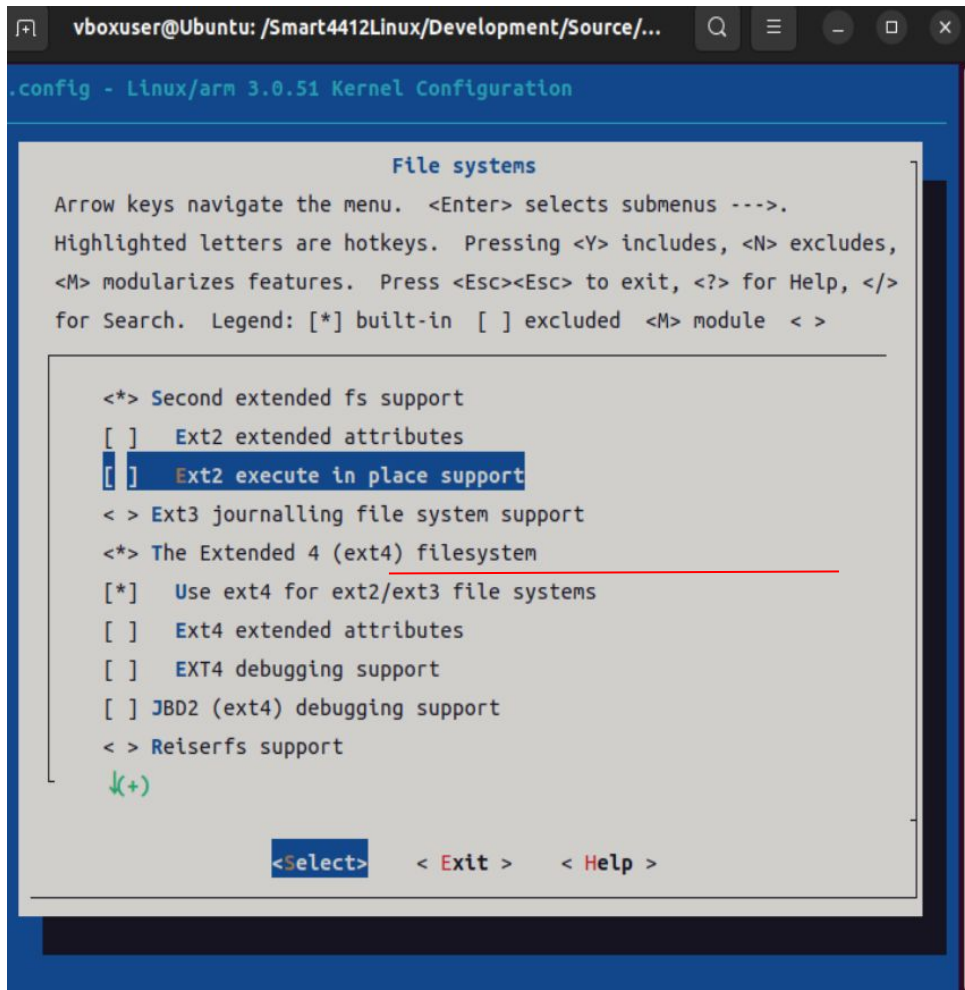
2. 파일 시스템 빌드 준비

- 1) 파일 시스템 빌드 과정을 전체적으로 설명하시오.
- 2) 커널 `configuration`에서 `ext4`가 선택되었는지 확인하시오.
- 3) 파일 시스템 설치 파일 저장 및 압축 해제

1) 파일 시스템 빌드 과정을 전체적으로 설명하시오.

1. 파일 시스템이 빌드된 디렉토리 생성
2. 파일 시스템의 소스 코드를 다운로드하고 압축을 푼다
3. 소스코드를 컴파일 전 **configure** 스크립트를 실행하여 파일 시스템이 빌드되는 환경을 설정한다.
4. **configure** 스크립트가 실행되면 **Makefile**이 생성되는데, 이 **Makefile**은 소스코드를 컴파일하고 파일 시스템을 빌드하는 데 필요한 명령을 정의
5. **Make** 명령어를 실행하여 파일시스템을 빌드한다.
6. 파일 시스템을 컴파일하고 빌드한 후에는 파일 시스템을 설치할 위치를 지정하여 **make install** 명령을 실행

2) 커널 configuration에서 ext4가 선택되었는지 확인하시오.



The screenshot shows a terminal window titled "vboxuser@Ubuntu: /Smart4412Linux/Development/Source/...". The main window is the "Linux/arm 3.0.51 Kernel Configuration" menu. The "File systems" section is expanded, showing a list of options. The option "Ext2 execute in place support" is highlighted with a blue background. Below it, "Ext3 journalling file system support" is shown. The "The Extended 4 (ext4) filesystem" option is underlined in red. Below this, "Use ext4 for ext2/ext3 file systems" is marked with an asterisk, indicating it is selected. Other options like "Ext4 extended attributes", "EXT4 debugging support", and "JBD2 (ext4) debugging support" are shown with empty brackets. At the bottom, there are navigation buttons: "<Select>", "< Exit >", and "< Help >".

```
vboxuser@Ubuntu: /Smart4412Linux/Development/Source/...  
Linux/arm 3.0.51 Kernel Configuration  
  
File systems  
Arrow keys navigate the menu. <Enter> selects submenus --->.  
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,  
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>  
for Search. Legend: [*] built-in [ ] excluded <M> module < >  
  
<*> Second extended fs support  
[ ] Ext2 extended attributes  
[ ] Ext2 execute in place support  
< > Ext3 journalling file system support  
<*> The Extended 4 (ext4) filesystem  
[*] Use ext4 for ext2/ext3 file systems  
[ ] Ext4 extended attributes  
[ ] EXT4 debugging support  
[ ] JBD2 (ext4) debugging support  
< > Reiserfs support  
  
(+)  
  
<Select> < Exit > < Help >
```

3) 파일 시스템 설치 파일 저장 및 압축 해제

```
root@Ubuntu: /Smart4412Linux/Development/Source/Files...
vboxuser@Ubuntu: /Smart4412Linux$ su
Password:
root@Ubuntu: /# cd Smart4412Linux
root@Ubuntu: /Smart4412Linux# cd Development/Source/filesystem
bash: cd: Development/Source/filesystem: No such file or directory
root@Ubuntu: /Smart4412Linux# cd Development/Source/
root@Ubuntu: /Smart4412Linux/Development/Source# ls
bootloader  Filesystem  kernel
root@Ubuntu: /Smart4412Linux/Development/Source# cd Filesystem
root@Ubuntu: /Smart4412Linux/Development/Source/Filesystem# ls
mkfs.sh  rootfs_H4412  rootfs_H4412.tgz
root@Ubuntu: /Smart4412Linux/Development/Source/Filesystem# tar -zxvf rootfs_H4412
.tgz
tar (child): rootfs_H4412.tgz: Cannot open: No such file or directory
tar (child): Error is not recoverable: exiting now
tar: Child returned status 2
tar: Error is not recoverable: exiting now
root@Ubuntu: /Smart4412Linux/Development/Source/Filesystem# tar -zxvf rootfs_H441
2.tgz
rootfs_H4412/
rootfs_H4412/sbin/
rootfs_H4412/sbin/mdev
rootfs_H4412/sbin/start-stop-daemon
```

3. 파일 시스템 이미지 빌드

1) 파일 시스템 빌드 과정에 따라서 빌드하여 `rootfs_ext4.img`를 생성하시오.

1) 파일 시스템 빌드 과정에 따라서 빌드하여 rootfs_ext4.img를 생성하시오.

```
rootfs_H4412/run
root@Ubuntu: /Smart4412Linux/Development/Source/Filesystem# chmod 777 mkfs.sh
root@Ubuntu: /Smart4412Linux/Development/Source/Filesystem# ls
mkfs.sh  rootfs_H4412  rootfs_H4412.tgz
root@Ubuntu: /Smart4412Linux/Development/Source/Filesystem# ./mkfs.sh
[ create rootfs_ext4.img ]
[ rootfs_H4412 size 252 MB ]
[ image size 315 MB ]
310+0 records in
310+0 records out
325058560 bytes (325 MB, 310 MiB) copied, 0.970077 s, 335 MB/s
losetup: rootfs_ext4.img: failed to set up loop device: Device or resource busy
[ mkfs.ext4 ]
mke2fs 1.46.5 (30-Dec-2021)
/dev/loop0 is mounted; will not make a filesystem here!
[ mount tmp_dir ]
mount: /Smart4412Linux/Development/Source/Filesystem/tmp_dir: WARNING: source write-protected, mounted read-only.
[ cp -a rootfs_H4412/* tmp_dir/ ]
cp: cannot overwrite non-directory 'tmp_dir/bin' with directory 'rootfs_H4412/bin'
cp: cannot create symbolic link 'tmp_dir/dev/log': Read-only file system
cp: cannot create symbolic link 'tmp_dir/dev/fd': Read-only file system
cp: cannot create directory 'tmp_dir/dev/mapper': Read-only file system
```

```
m
cp: preserving times for 'tmp_dir/usr/share/terminfo': Read-only file system
cp: cannot create directory 'tmp_dir/usr/share/gtk-2.0': Read-only file system
cp: preserving times for 'tmp_dir/usr/share': Read-only file system
cp: preserving times for 'tmp_dir/usr': Read-only file system
cp: cannot overwrite directory 'tmp_dir/var/spool' with non-directory
cp: cannot overwrite directory 'tmp_dir/var/log' with non-directory
cp: cannot overwrite directory 'tmp_dir/var/tmp' with non-directory
cp: cannot overwrite directory 'tmp_dir/var/lib/misc' with non-directory
cp: cannot create symbolic link 'tmp_dir/var/lib/pcmcia': Read-only file system
cp: cannot create directory 'tmp_dir/var/lib/alsa': Read-only file system
cp: cannot overwrite directory 'tmp_dir/var/lib/dbus' with non-directory
cp: preserving times for 'tmp_dir/var/lib': Read-only file system
cp: cannot create symbolic link 'tmp_dir/var/pcmcia': Read-only file system
cp: cannot create directory 'tmp_dir/var/www': Read-only file system
cp: cannot overwrite directory 'tmp_dir/var/cache' with non-directory
cp: cannot remove 'tmp_dir/var/lock': Read-only file system
cp: cannot remove 'tmp_dir/var/run': Read-only file system
cp: preserving times for 'tmp_dir/var': Read-only file system
[ umount tmp_dir ]
[ rootfs_ext4.img done. ]
root@Ubuntu: /Smart4412Linux/Development/Source/Filesystem# ls
mkfs.sh  rootfs_ext4.img  rootfs_H4412  rootfs_H4412.tgz
root@Ubuntu: /Smart4412Linux/Development/Source/Filesystem#
```