
Linux File System

JEONBUK NATIONAL UNIVERSITY



Contents

A.Linux File System

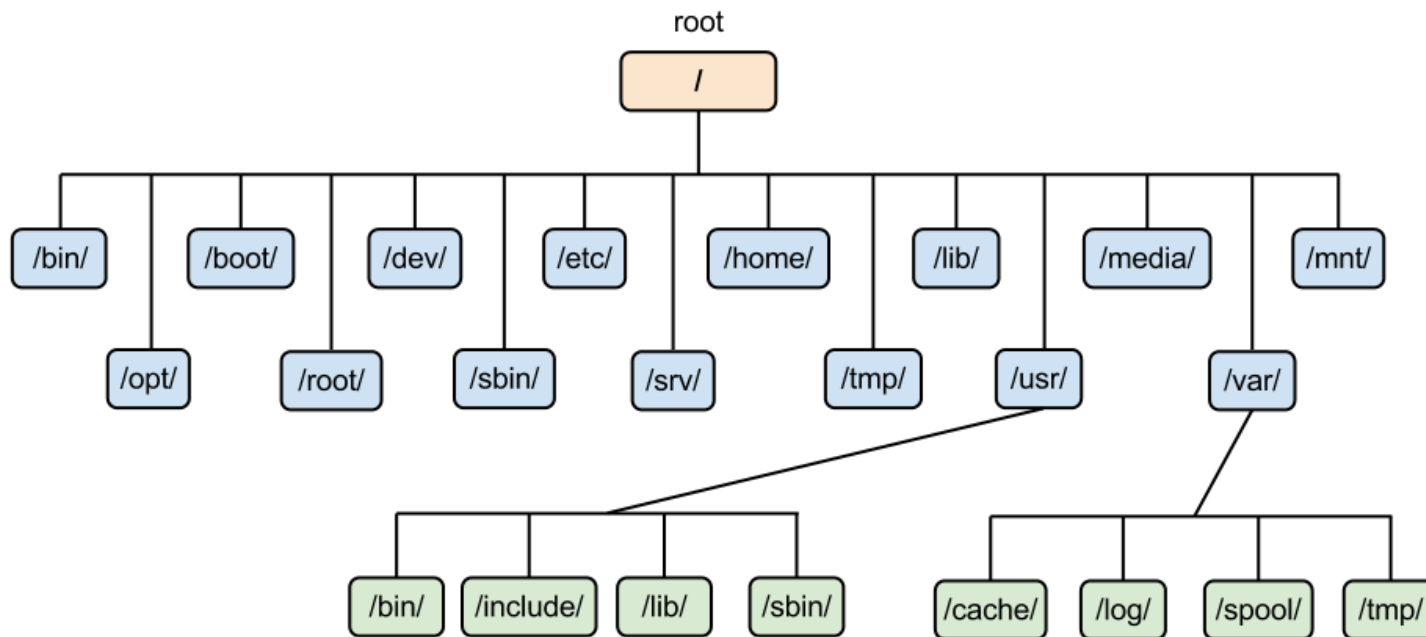
B.Registry in Linux

1 Linux File System

JEONBUK NATIONAL UNIVERSITY

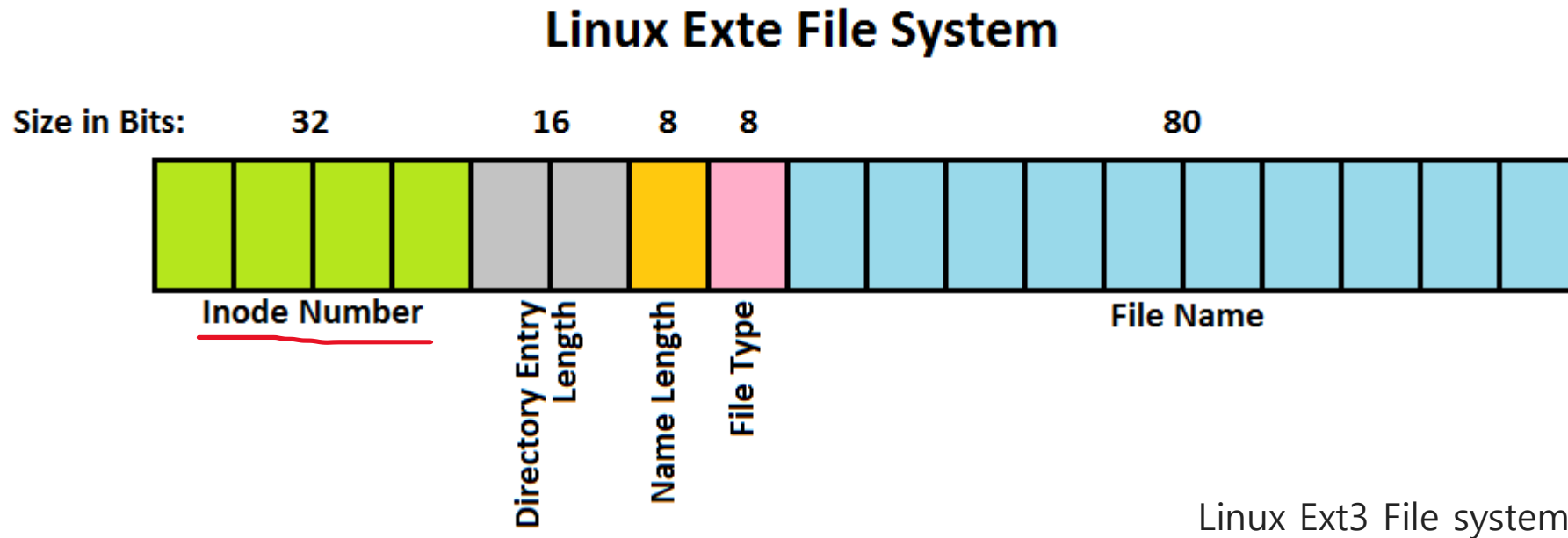
- **File** : 데이터를 담는 그릇
- **File System** : File을 관리하는 시스템
- File System은 파일을 관리할 정보가 필요

- Directory : 파일을 편하게 관리하기 위한 도구

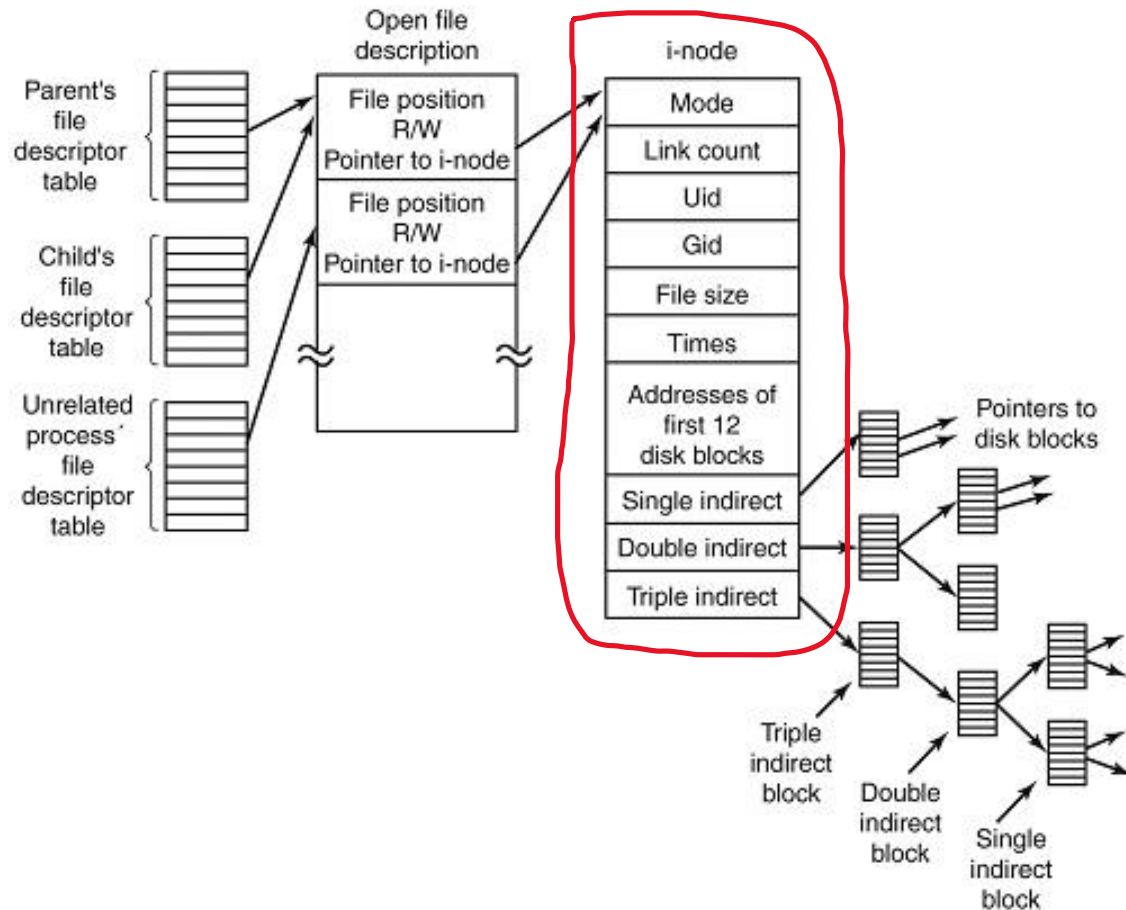


Directory는 트리 구조

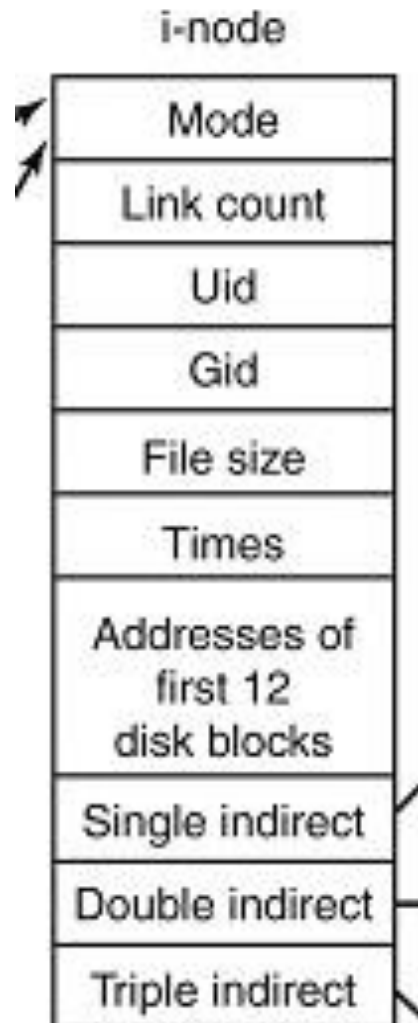
- Directory Entry: directory를 표현하는 자료구조



- I-Node(Index Node) : 파일에 대한 정보(meta data) 를 가진 일종의 데이터



파일이나 디렉토리는
고유한 i-node를 가지고 있다



- File mode(permission)
- Number of links to the file
- UID of the owner
- GID of the owner
- Size of the file
- Actual number of block that the file uses
- Time last modified
- Time last accessed
- Time last changed
- Block pointer



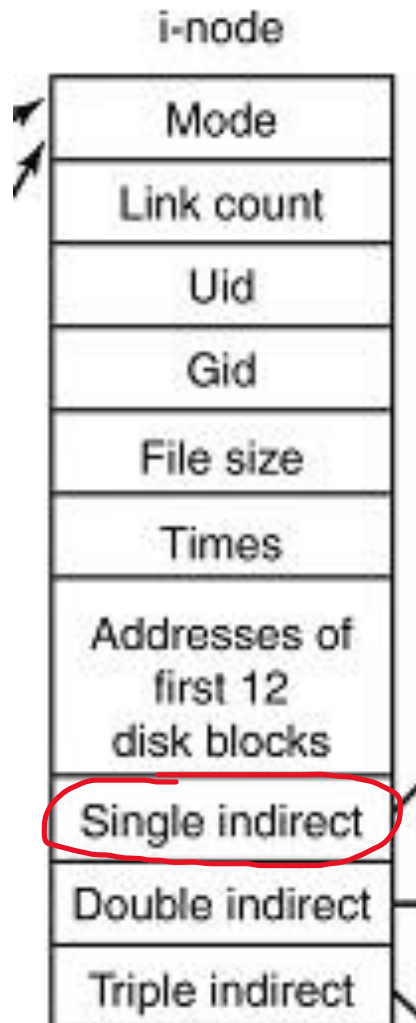
■ Direct Block Pointers

In an ext2 file system an inode consists of only 15 block pointers. The first 12 block pointers are called as Direct Block pointers.

Which means that these pointers point to the address of the blocks containing the data of the file.

12 Block pointers can point to 12 data blocks. So in total the Direct Block pointers can address only 48KB($12 * 4KB$) of data.

Which means if the file is only of 48KB or below in size, then inode itself can address all the blocks containing the data of the file.

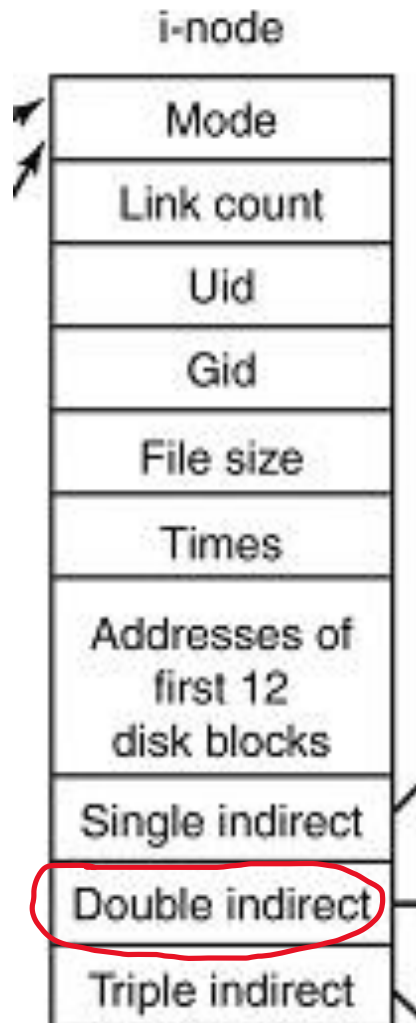


■ Indirect Block Pointers

whenever the size of the data goes above 48KB (by considering the block size as 4KB), the 13th pointer in the inode will point to the very next block after the data (adjacent block after 48KB of data), which in turn will point to the next block address where data is to be copied.

Now as we have taken our block size as 4KB, the indirect block pointer can point to 1024 blocks containing data (by taking the size of a block pointer as 4 bytes, one 4KB block can point to 1024 blocks because $4 \text{ bytes} * 1024 = 4\text{KB}$).

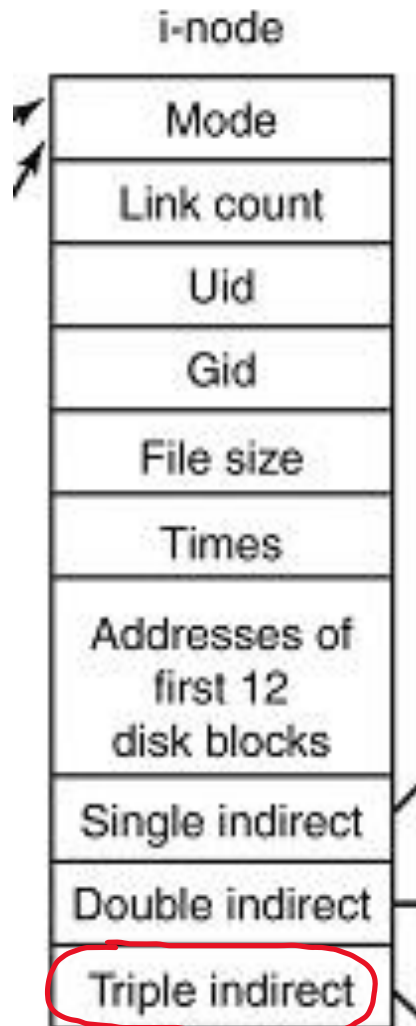
which means an indirect block pointer can address, up to 4MB of data (4 bytes of block pointer in 4KB block, can point and address 1024 number of 4K blocks which makes the data size of 4M).



■ Double Indirect Block Pointers

Now if the size of the file is above 4MB + 48K then the inode will start using Double Indirect Block Pointers, to address data blocks. Double Indirect Block pointer in an inode will point to the block that comes just after 4M + 48K data, which intern will point to the blocks where the data is stored.

Double Indirect block pointer also is inside a 4K block as every blocks are 4K, Now block pointers are 4 bytes in size, as mentioned previously, so Double indirect block pointer can address 1024 Indirect Block pointers(which means $1024 * 4M = 4G$). So with the help of a double indirect Block Pointer the size of the data can go upto 4G.



■ Triple Indirect Block Pointers

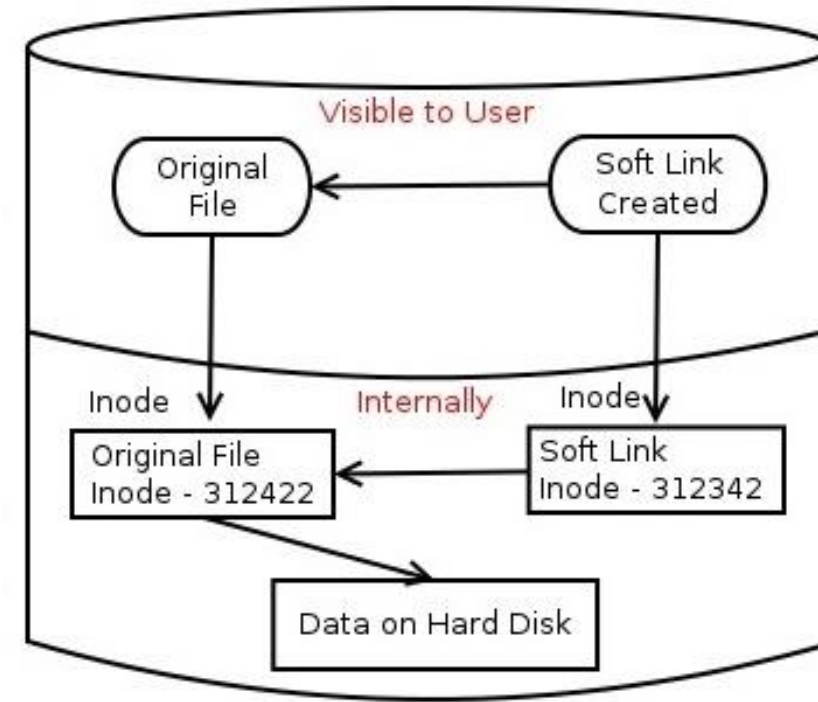
Now this triple Indirect Block Pointers can address up to $4G * 1024 = 4TB$, of file size.

The fifteenth block pointer in the inode will point to the block just after the 4G of data, which in turn will point to 1024 Double Indirect Block Pointers.

■ Symbolic Link

심볼릭 링크는 원본 파일의 inode를 가리키는 정보가 있는 파일.

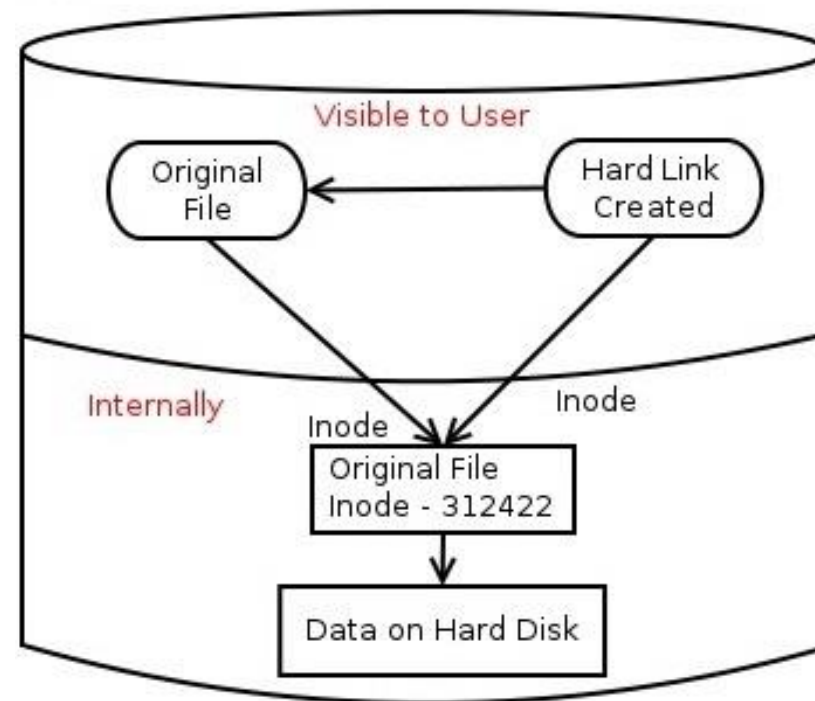
이 inode는 하드디스크의 데이터를 가리킨다.

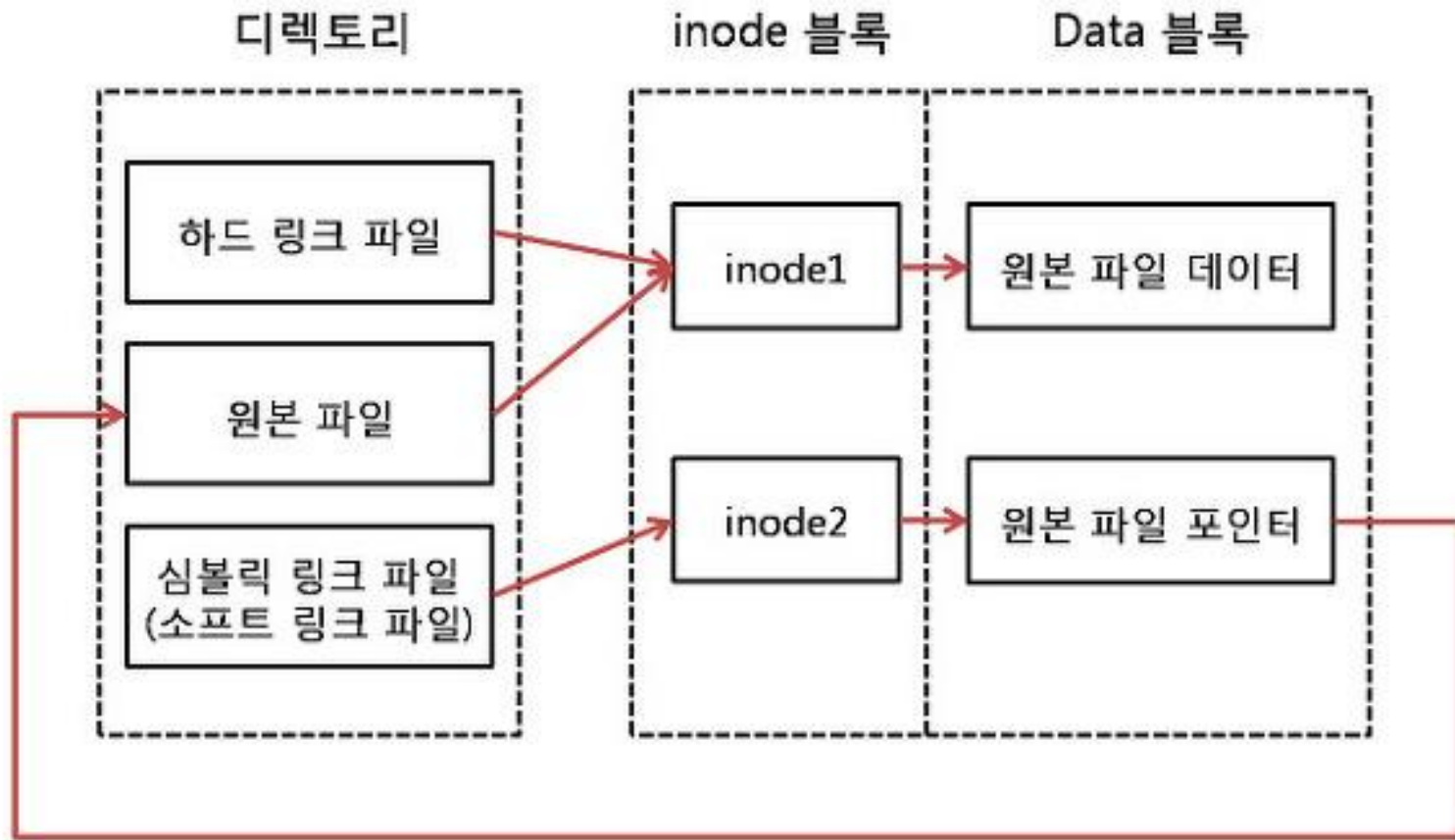


■ Hard Link

하드 링크는 동일한 inode를 복사해 사용

복사되는 파일의 inode에서 링크 카운트가 하나 증가





```
root@server: ~/linktest
root@server:~/바탕화면# cd
root@server:~# mkdir linktest
root@server:~# cd linktest/
root@server:~/linktest# pwd
/root/linktest
root@server:~/linktest# nano basefile
root@server:~/linktest# cat basefile
원본 파일 입니다...
root@server:~/linktest#
```

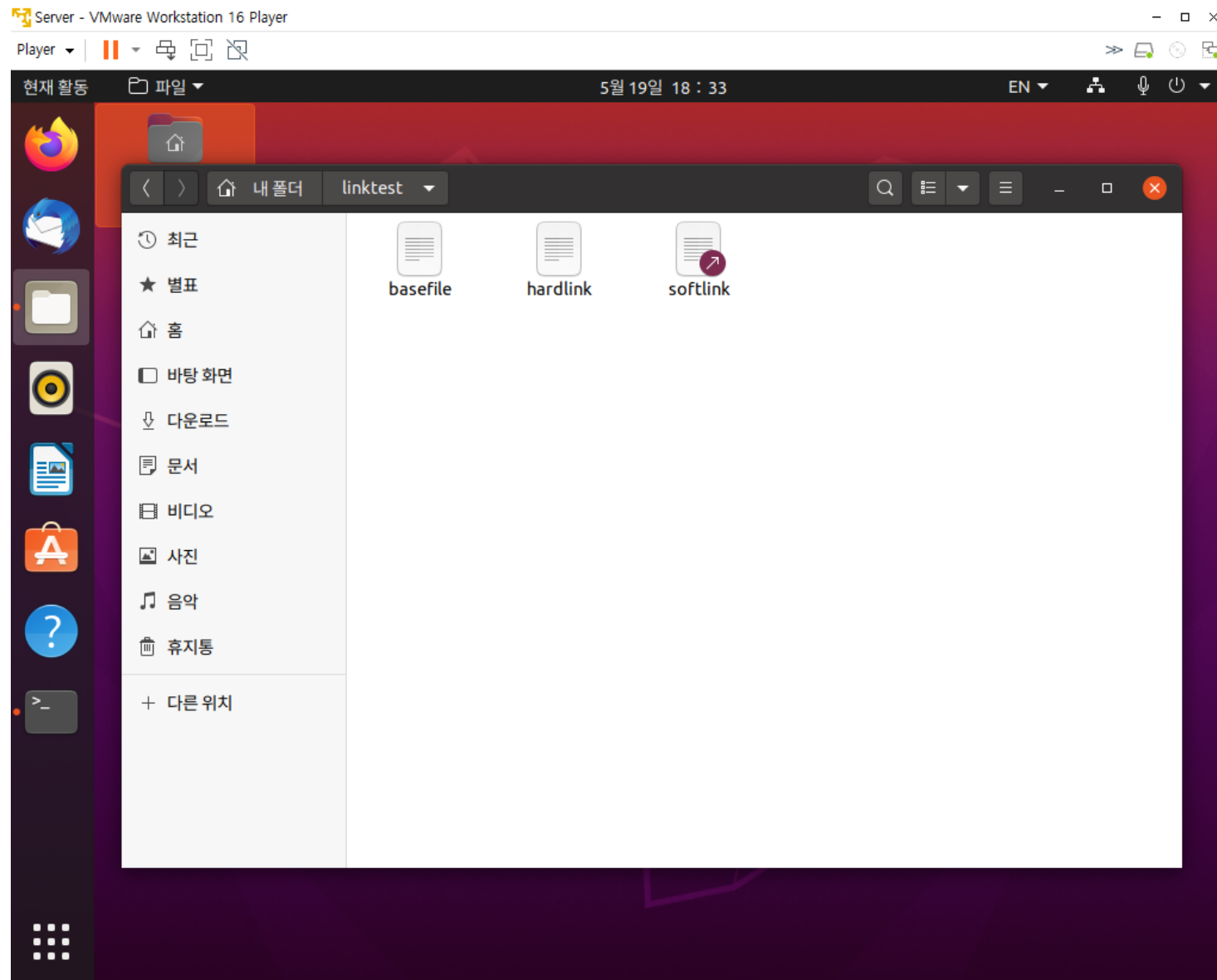


```

root@server: ~/linktest
root@server:~/바탕화면# cd
root@server:~# mkdir linktest
root@server:~# cd linktest/
root@server:~/linktest# pwd
/root/linktest
root@server:~/linktest# nano basefile
root@server:~/linktest# cat basefile
원본 파일 입니다...
root@server:~/linktest# ln basefile hardlink
root@server:~/linktest# ln -s basefile softlink
root@server:~/linktest# ls -il
합계 8
262267 -rw-r--r-- 2 root root 27 5월 19 18:23 basefile
262267 -rw-r--r-- 2 root root 27 5월 19 18:23 hardlink
262192 lrwxrwxrwx 1 root root 8 5월 19 18:25 softlink -> basefile
root@server:~/linktest#
    
```

```

root@server: ~/linktest
root@server:~/바탕화면# cd
root@server:~# mkdir linktest
root@server:~# cd linktest/
root@server:~/linktest# pwd
/root/linktest
root@server:~/linktest# nano basefile
root@server:~/linktest# cat basefile
원본 파일 입니다...
root@server:~/linktest# ln basefile hardlink
root@server:~/linktest# ln -s basefile softlink
root@server:~/linktest# ls -il
합계 8
262267 -rw-r--r-- 2 root root 27  5월 19 18:23 basefile
262267 -rw-r--r-- 2 root root 27  5월 19 18:23 hardlink
262192 lrwxrwxrwx 1 root root  8  5월 19 18:25 softlink -> basefile
root@server:~/linktest# mv basefile ../
root@server:~/linktest# ls
hardlink softlink
root@server:~/linktest# cat hardlink
원본 파일 입니다...
root@server:~/linktest# cat softlink
cat: softlink: 그런 파일이나 디렉터리가 없습니다
root@server:~/linktest#
    
```



구분	심볼릭 링크	하드 링크
생성 명령어	ln -s [원본 파일명] [링크 파일명]	ln [원본 파일명] [링크 파일명]
생성 종류	파일과 디렉토리 모두 생성	파일만 생성
링크 기능	파일 또는 디렉토리 이름에 대한 링크를 가리킴	원본 파일에 대한 참조 또는 포인터
원본 파일 삭제할 경우	액세스 불가능	액세스 가능
inode 번호	<u>다른</u> inode 번호	<u>동일한</u> inode 번호
다른 파티션 링크 여부	다른 파티션에 링크 가능	다른 파티션에 링크 불가능
특징	<ul style="list-style-type: none"> - 데이터 접근 시, 원본 i-node를 경유한다. - 디렉터리도 가능하다. 	<ul style="list-style-type: none"> - i-node로 바로 데이터에 접근한다. - 디렉터리는 지원하지 않음

■ Symbolic Link

1. 파일 시스템에 링크할 경우 사용
2. 디렉토리를 링크할 경우 사용

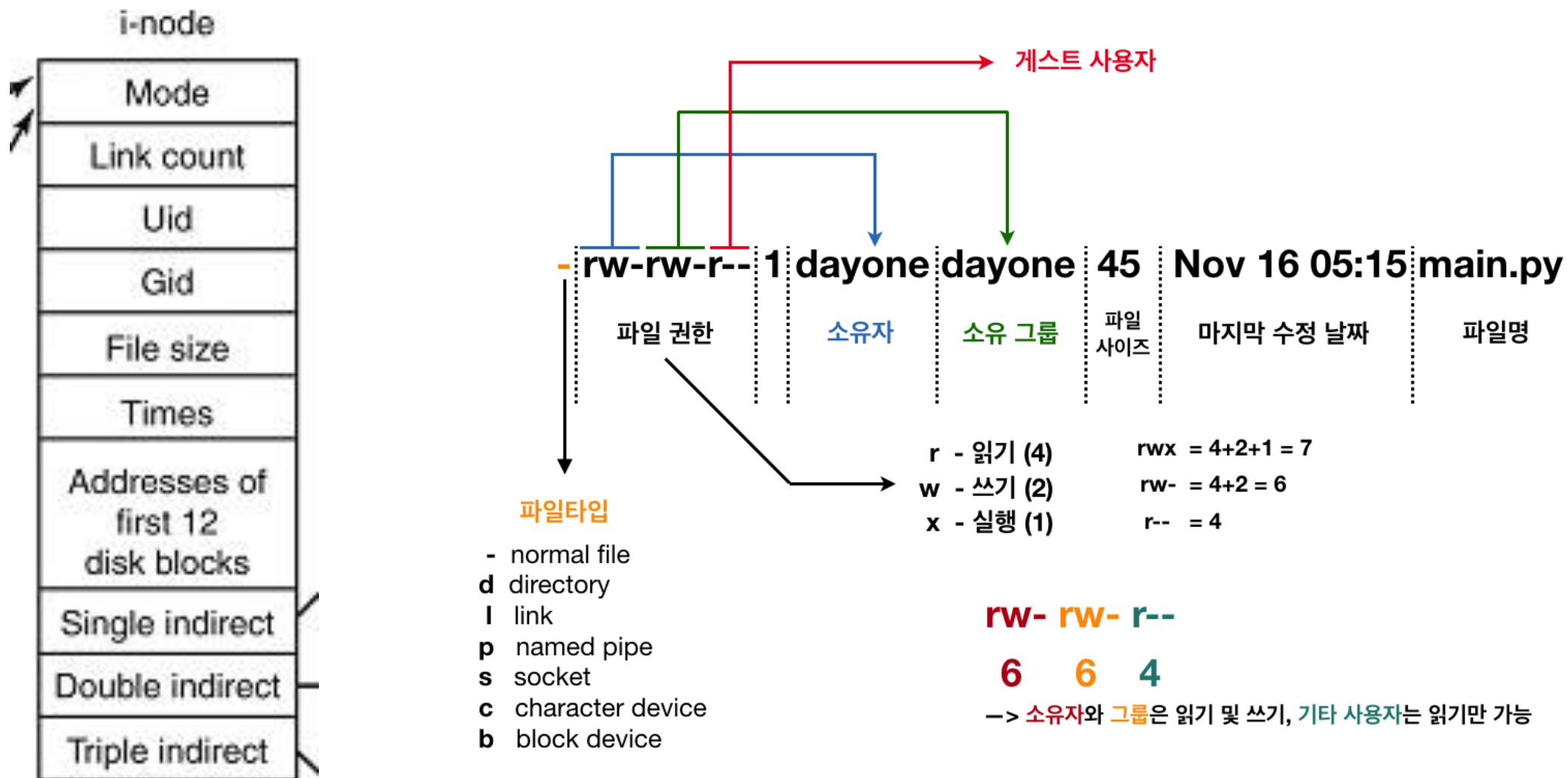
■ Hard Link

1. 자원을 공유하면서 데이터를 안전하게 관리하고자 할 때 주로 사용

■ Permission

특정 파일이나 디렉터리에 대하여 읽기, 기록하기, 삭제하기 등의 권한을 설정해 놓은 것으로 다중 사용자 운영체제에서 파일의 접근권한과 보호 등을 위하여 반드시 필요한 것

1. User
2. Group
3. Others

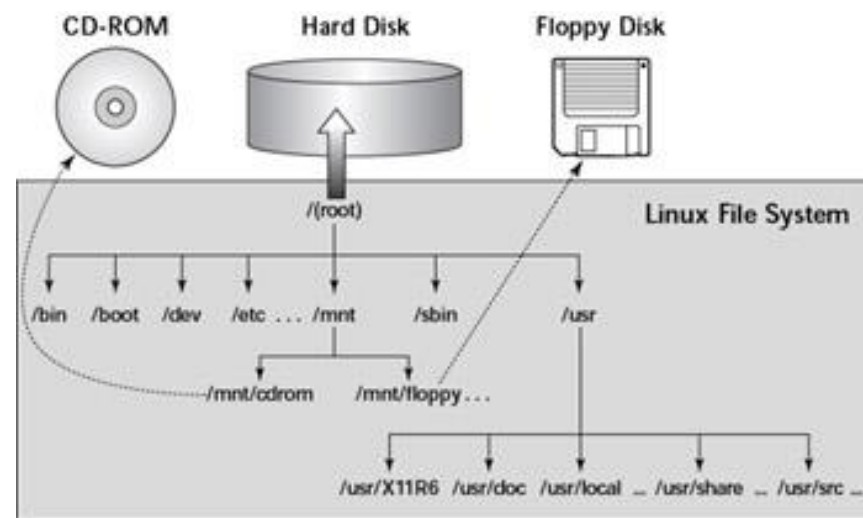


	<div> <div>u</div> <div>g</div> <div>o</div> </div> <div>754</div>								
access	r	w	x	r	w	x	r	w	x
binary	4	2	1	4	2	1	4	2	1
enabled	1	1	1	1	0	1	1	0	0
result	4	2	1	4	0	1	4	0	0
total	7			5			4		

■ Mount

Mount란 하드디스크 파티션, CD/DVD, USB 메모리 등 물리적인 장치를 특정 위치에 연결시키는 과정

윈도우 운영체제에서는 PnP 기능에 의해 자동으로 주변 장치를 인식하기 때문에 직접 마운트 작업을 하지 않아도 되지만 리눅스 환경에서는 각각의 파티션들을 마운트 해주어야 함



■ File Allocation

디스크에 파일을 저장하기 위해 공간을 관리해야 함

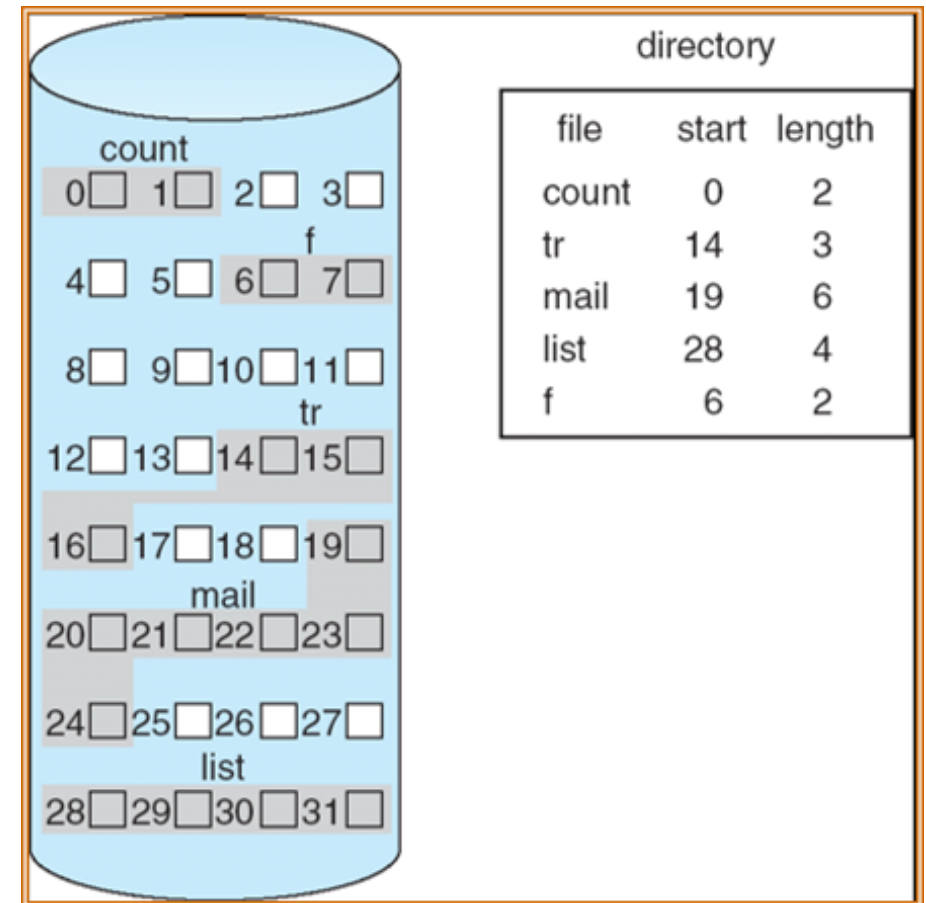
1. Contiguous Allocation
2. Linked Allocation
3. FAT
4. Indexed Allocation
5. Multilevel Index

■ Contiguous Allocation

디스크에 연속적으로 file block을 할당

특정 파일을 append 할 때 여유공간이 없다면 충분한 공간이 있는 위치로 이동시킨 뒤 확장

Block 들을 옮기고 지우는 방식은 copy & paste를 하게 되는데, 그러면 필연적으로 disk access 횟수가 늘어나게 되므로 매우 느림



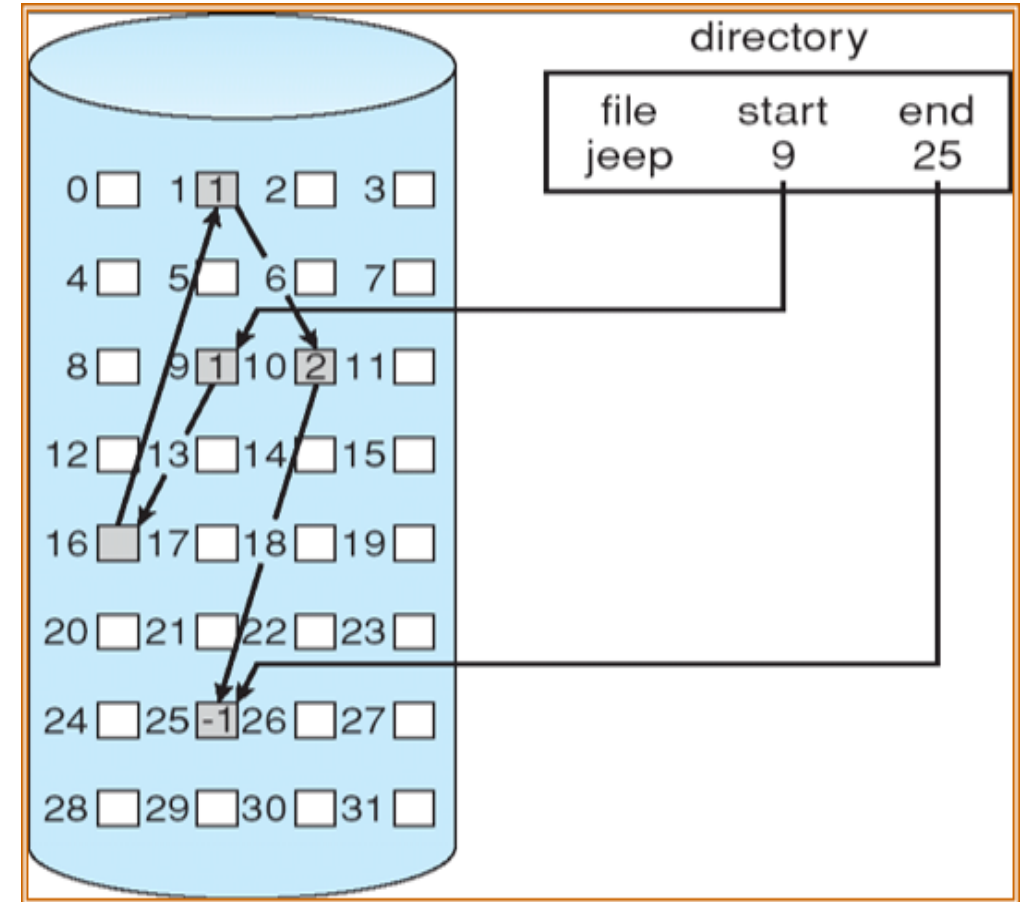
▪ Linked Allocation

Contiguous Allocation의 단점을 보완

Data block이 디스크 상에서 흩어져 있지만 link를 통해 연결

마지막 블록의 link에는 끝임을 표시하는 nil(-1)이 저장

Linked list의 단점인 sequential access 존재



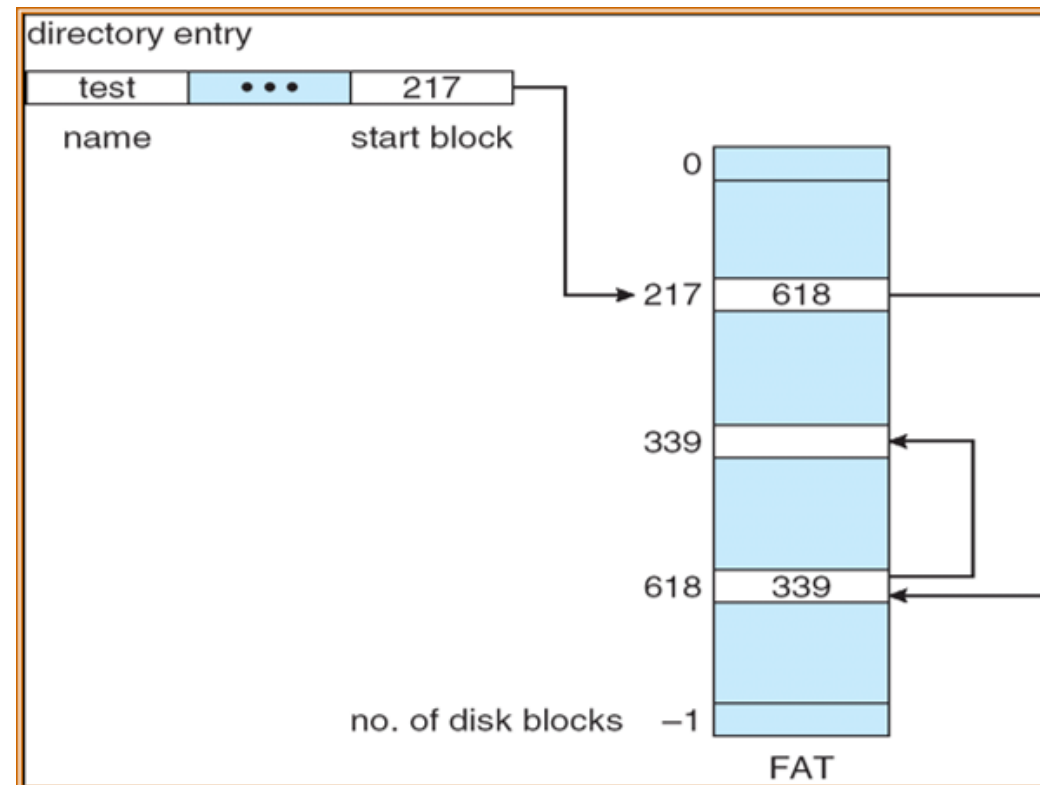
■ FAT(File Allocation Table)

특정 파일의 link정보를 저장

Link 정보를 따로 저장하므로, disk의 추가 용량이 필요 없음

시작 block주소만 알고 있다면 FAT를 탐색하여 이후 block의 주소를 알 수 있다

하지만 table을 위한 공간이 따로 필요한데, 이를 해결하기 위해 main memory의 cache에 해당 table을 저장

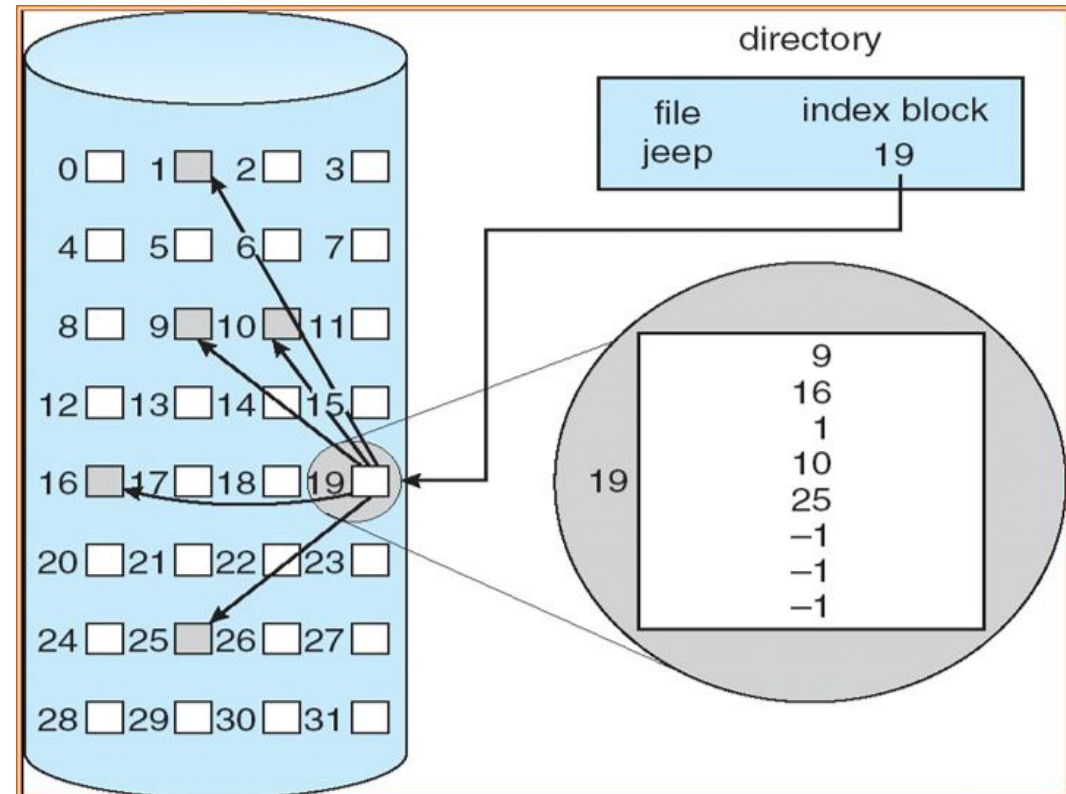


Indexed Allocation

하나의 block이 table 역할을 함

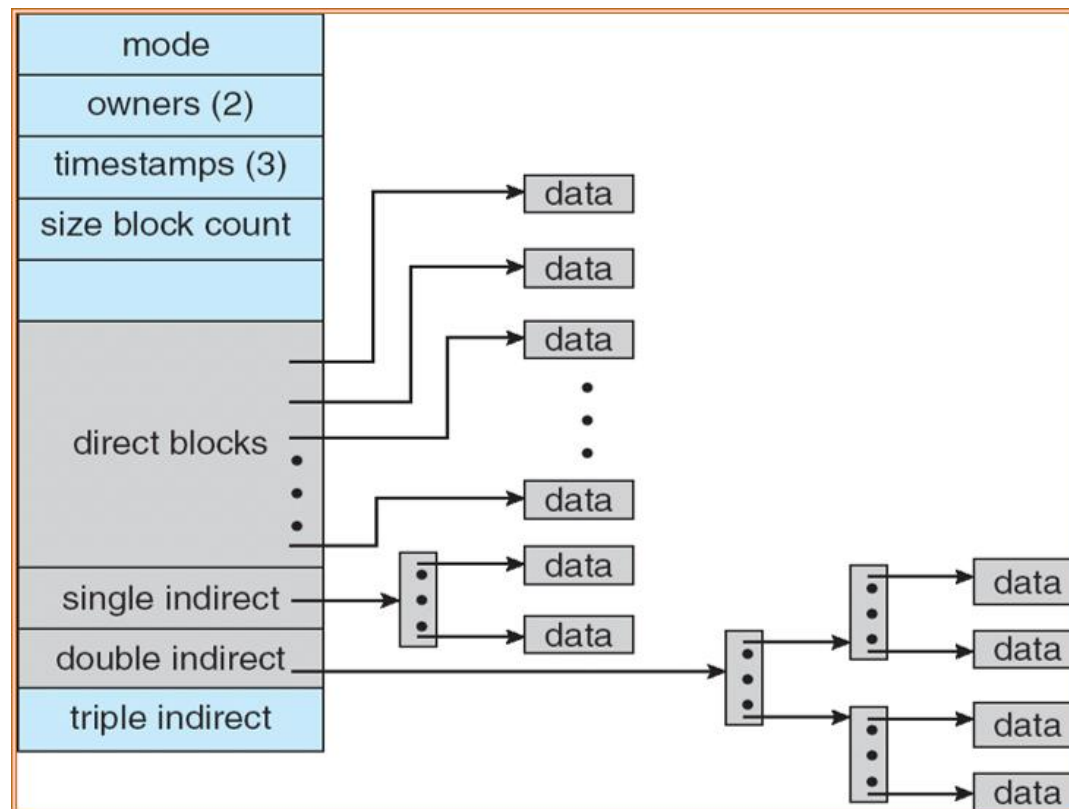
file data block이 한 두개 정도인 경우라도 무조건 index block을 할당해야 하므로, 공간 낭비 발생

큰 용량의 file을 저장하기 힘들다는 단점 존재



■ Multilevel Index

12개의 direct block과 하나의 single, double, triple indirect block을 가지는 방식



2 Registry In Linux

JEONBUK NATIONAL UNIVERSITY

■ Window Registry

서비스, 구성요소, 각종 프로그램 처럼 window 전반에 걸쳐 사용되는 설정 그리고 구성을 포함하는 데이터베이스

■ Registry in Linux

Window registry같은 데이터베이스는 없음

구성은 대부분 텍스트 파일에 보관

1. /proc
2. /etc

■ /proc

프로세스, 하드웨어, 시스템 정보를 가지는 디렉토리

```

root@server: /proc
root@server:/proc# ls
1      118   1384   1537   207   237   269   467   862   98      mdstat
10     119   1388   1540   208   238   270   469   863   982     meminfo
100    12    1397   1543   209   239   271   475   864   99      misc
1007   120   14    1545   21    24    272   479   868   acpi     modules
1008   121   1400   1556   210   240   273   490   87    buddyinfo
101    1215  1407   1582   211   241   274   495   870   bus      mounts
1013   122   1418   16    212   242   275   496   873   cgroups  mtrr
1014   123   1420   1625   213   243   276   5    874   cmdline net
102    1233  1424   1640   214   244   277   500   88    consoles pagetypeinfo
103    1236  1431   1692   215   245   278   551   887   cpuinfo  partitions
1035   124   1437   17    216   246   279   552   889   crypto   pressure
1037   1241  1441   1721   217   247   280   554   89    devices  sched_debug
104    1243  1452   18    218   248   281   555   898   diskstats schedstat
1044   1244  1458   1802   219   249   282   557   9    dma       scsi
1047   1248  1468   1821   22    250   283   6    90    driver    self
105    125    1472   1831   220   251   284   690   901   execdomains slabinfo
1052   1253  1475   19    221   252   285   7    902   fb         softirqs
1058   1257  1480   192    222   253   286   70   903   filesystems stat
106    1282  1481   193    223   254   3    71   908   fs        swaps
107    1287  1488   195    224   255   316   72   91    interrupts sys
108    1291  1489   196    225   256   317   73   910   iomem     sysrq-trigger
109    1295  1490   197    226   257   356   74   919   ioports   sysvipc
11    13    1491   198    227   258   382   75   92    irq        thread-self
110   1302  1499   199    228   259   385   76   929   kallsyms  timer_list
111   1307   15    2    229   260   397   77   93    kcore     tty
112   1315   150    20    23    261   4    78   94    key-users uptime
1127   1321  1502   200    230   262   401   79   948   keys       version
113    1328  1508   201    231   263   402   8    949   kmsg       version_signature
1131   1335  1515   202    232   264   425   81   95    kpagecgroup vmallocinfo
114    134    1524   203    233   265   433   82   957   kpagecount vmstat
115   1351  1528   204    234   266   435   84   958   kpageflags zoneinfo
116   1367  1529   205    235   267   438   85   96    loadavg
117    137    1533   206    236   268   450   86   97    locks
root@server:/proc#

```

```

Server - VMware Workstation 16 Player
Player | | | | |
5월 20일 10 : 49
root@server: /proc
root@server:/proc# cat cpuinfo
processor       : 0
vendor_id      : AuthenticAMD
cpu family     : 23
model          : 8
model name     : AMD Ryzen 5 2600 Six-Core Processor
stepping       : 2
microcode      : 0x8008206
cpu MHz        : 3399.998
cache size     : 512 KB
physical id    : 0
siblings       : 1
core id        : 0
cpu cores      : 1
apicid         : 0
initial apicid : 0
fpu            : yes
fpu_exception  : yes
cpuid level    : 13
wp             : yes
flags          : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fx
sr sse sse2 syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm constant_tsc rep_good nopl tsc_reliable nonstop
_tsc cpuid extd_apicid pni pclmulqdq sse3 fma cx16 sse4_1 sse4_2 x2apic movbe popcnt aes xsave avx f16c
rdrand hypervisor lahf_lm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw topoext ssbd ibpb
vmcall fsgsbase bmi1 avx2 smep bmi2 rdseed adx smap clflushopt sha_ni xsaveopt xsavec xgetbv1 xsaves c
lzero arat overflow_recov succor
bugs           : fxsave_leak sysret_ss_attrs null_seg spectre_v1 spectre_v2 spec_store_bypass
bogomips       : 6799.99
TLB size       : 2560 4K pages
clflush size   : 64
cache_alignment : 64
address sizes   : 45 bits physical, 48 bits virtual
power management:
root@server:/proc#

```

■ /etc, /usr/etc

시스템의 부팅, 쉼다운 시에 필요한 파일들과 시스템의 전반에 걸친 설정 파일들 및 초기 스크립트 파일 저장

```
root@server: /etc
root@server:/etc# ls
NetworkManager  fprintd.conf  logrotate.conf  rmt
PackageKit      fstab          logrotate.d     rpc
UPower          fuse.conf     lsb-release     rsyslog.conf
X11             fwupd         ltrace.conf     rsyslog.d
acpi            gai.conf      machine-id      rygel.conf
adduser.conf    gamemode.ini  magic           sane.d
alsa            gdb           magic.mime      security
alternatives    gdm3          mailcap         selinux
anacrontab      geoclue       mailcap.order  sensors.d
apg.conf        ghostscript   manpath.config  sensors3.conf
apm             glvnd         mime.types     services
apparmor        gnome         mke2fs.conf    sgml
apparmor.d      groff         modprobe.d     shadow
appport         group         modules        shadow-
appstream.conf  grub.d        modules-load.d  shells
apt             gshadow       mtab           skel
avahi           gshadow-     mttools.conf   snmp
bash.bashrc     gss           mysql          speech-dispatcher
bash_completion gss           nanorc         ssh
bash_completion.d gtk-2.0      netplan        ssl
bindresvport.blacklist gtk-3.0     network        subgid
binfmt.d        hdparm.conf  networkd-dispatcher subuid
bluetooth       host.conf    networks       subuid-
brlapi.key      hostid       newt            sudoers
brltty          hostname     nsswitch.conf  sudoers.d
brltty.conf     hosts        openvpn        sysctl.conf
ca-certificates hosts.allow  os-release     sysctl.d
ca-certificates.conf hosts.deny  pam.conf      systemd
calendar        ifplugd     papersize      termInfo
chatscripts      console-setup init          thermald
cracklib         cron.d       initt.d        thunderbird
cron             cron.daily  initramfs-tools timezone
cron.hourly      cron.monthly inputrc        tmpfiles.d
cron.monthly     cron.weekly inserv.conf.d ubuntu-advantage
cron.weekly      cron.weekly iproute2       ucf.conf
crontab          issue       pm             udev
fprintd.conf     issue.net   rpm2cpio.conf uidisks2
```

```
root@server: /etc
root@server:/etc# cat passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
messagebus:x:100:103::/nonexistent:/usr/sbin/nologin
syslog:x:101:107::/home/syslog:/usr/sbin/nologin
_apt:x:102:65534::/nonexistent:/usr/sbin/nologin
_apt:x:103:108:TPM software stack,,,:/var/lib/tpm:/bin/false
uidd:x:104:111::/run/uidd:/usr/sbin/nologin
cpdump:x:105:112::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:106:113:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:107:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
tkit:x:108:114:RealtimeKit,,,:/proc:/usr/sbin/nologin
dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
cups-pk-helper:x:110:117:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
speech-dispatcher:x:111:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
avahi:x:112:118:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
ternoops:x:113:65534:Kernel Oops Tracking Daemon,,,:/usr/sbin/nologin
saned:x:114:120::/var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:115:121:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
hplip:x:116:7:HPLIP system user,,,:/run/hplip:/bin/false
thpopsie:x:117:122::/nonexistent:/bin/false
```