

---

## 제 2 부

---

# AIX Administration

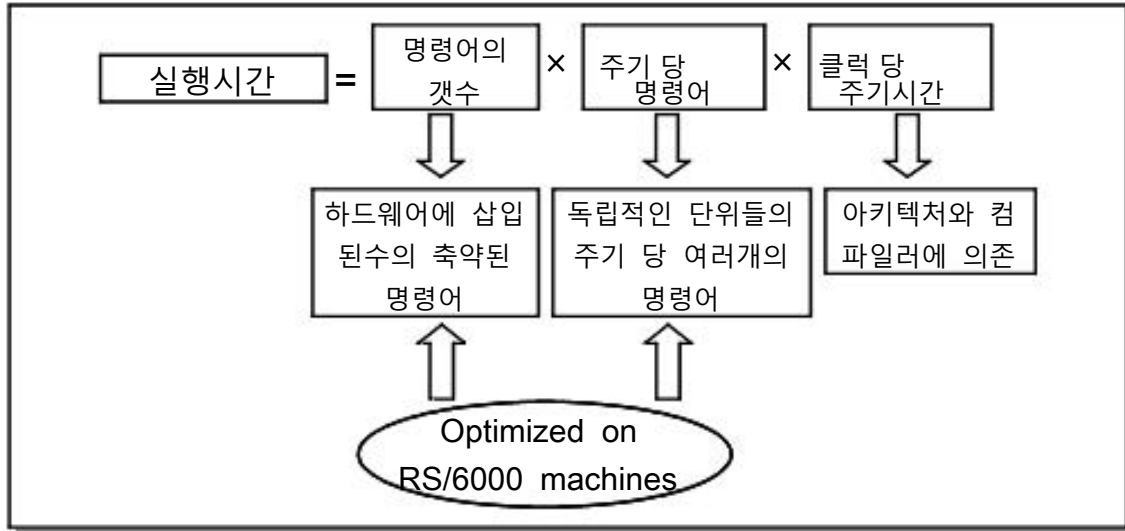
## Unit 1.Introduction to System P Administration

---

축소형 명령어 컴퓨터(RISC)의 기술이란?  
시스템 관리자의 목적  
시스템 관리자가 하는 일?  
시스템 관리자 가지는 권한?

## 축소 명령형 컴퓨터(RISC)의 기술이란?

### ▣ 프로세서 소요시간



### ▣ 프로세서들의 목적 :

- ▷ 하드웨어의 대부분 실행시간을 구현
- ▷ 주기당 여러 가지 명령어를 허용
- ▷ 하드웨어와 소프트웨어 사이에 시너지를 제공

### Notes :

- 실행 시간 : 프로그램 실행 시간은 전체적인 성능 또는 명령어들을 실행하는데 프로세서가 소요한 시간의 측정이다
- 명령어의 개수 : RISC 프로세서 아키텍처는 하드웨어에 삽입된 축약된수의 명령어들을 가진다. 컴파일러는 이러한 간결한 명령어 집합의 장점을 활용하도록 개발되었다.
- 주기 당 명령어 : POWER 아키텍처 내의 독립적인 단위들은 주기 당 여러 개의 명령어를 허용. POWER 칩은 분리된 기능 단위에 분기와 같은작업을 할당함으로써 여러 개의 명령어를 실행한다.
- 클럭 주기 당 시간 : 클럭 주기에 소요되는 시간은 아키텍처와 컴파일러에 의존합니다. AIX 컴파일러는 POWER 칩의 속도에 영향을 받는다.

※ Architecture : 내부구조, 레지스터, 기억 장치, 입출력 장치 등 컴퓨터의 기계적인 것을 나타냄

## 시스템 관리의 목적

---

- ▣ 시스템 관리의 의무
  - ▣ 사용자 또는 고객 요구사항
  - ▣ 네트워크 관리
- 

### Notes :

시스템 관리의 목표는 다른 모든 것의 관리 목표와도 정확하게 일치해야 한다. 사용자는 최소 비용으로 사용 가능한 자원으로부터 최대의 효과를 낼 수 있도록 시간과 자원을 효과적으로 관리해야 한다.

사용자는 또한 사용자의 비즈니스 요구사항에 대해 계획을 세워야 한다. 시스템 관리의 가장 중요한 목표는 회사 자원을 전사적으로 공평하고 적합하게 분배하고, 이들 자원에 대한 침입이나 장애로부터의 안전을 보장하는 것이다.

---

## 시스템 관리자가 하는 일?

---

- 이전설치 계획 :
  - ▷ 사용자 계정/그룹
  - ▷ 기억 할당/페이징 공간
  - ▷ 서브시스템(printing, networks...)
  - ▷ 표준 명칭 규정
- 하드웨어의 설정과 인스톨
- 소프트웨어의 설정
- 네트워크의 설정
- 시스템 백업
- 서버시스템 구축
- 시스템관리 지원(예:디스크공간)
- 모니터링 성능
- 용량 계획
- 제품의 라이선스관리

---

### Notes :

시스템이 장애를 겪게 될 때까지는 아무도 좋은 시스템 관리자를 필요로 하지 않는다. 시스템 장애에 대한 최상의 방어는 지식, 도구 및 계획이다.

시스템이 장애를 겪게 되면 시스템 관리자는 신속하게 시스템을 온라인 상태로 되돌리고 데이터 무결성을 보호한다.

시스템은 생산성, 확실성 및 성능을 극대화해야 하며,AIX는 유지보수를 위해 시스템을 종료하지 않고도 온라인 상태에서 디스크 공간을 동적으로 관리할 수 있게 하는 논리 볼륨 관리자(LVM)와 같은 기능으로 사용자에게 도움을 준다.

## 시스템 관리자의 권한?

### ■ 루트유저로서 유의사항.

- ▷ 항상 파일 권한은 제한한다.
- ▷ 함부로 루트로 로그인 하는 것은 피하도록 한다.
- ▷ 루트 패스워드는 안전하게 유지 해야한다.

### ■ 몇몇의 직무는 시스템의 안전과 printq와 같은 특정 그룹에서의 다른 사용자의 의해 수행 될 수 있다.

### ■ 아래의 su 명령어를 통해 루트사용자로 전환 한다.

`# su root` : 일반유저의 환경변수를 사용함.

`# su - root` : root 유저의 환경변수를 사용함.

## Notes :

### ■ 시스템 관리 작업에 대한 권한 부여

시스템 관리 작업은 많은 사용자에게 영향을 미칠 수 있으며 파장이 큰 결과를 낳을 수 있다. 필수 전문 기술을 갖춘 권한이 있는 사용자 만이 저장장치 할당 및 호스트 시스템 재구성과 같은 작업을 수행해야 하며, AIX 보안 권한은 특권이 있는 사용자에 대한 관리 작업의 수행을 제한한다. 특권이 있는 사용자 레벨 중 하나는 루트이다. 루트에게는 무제한의 액세스가 주어져서 시스템에 있는 어떤 것에도 액세스하여 변경할 수 있다.방금 검토한 것과 같은 대부분의 관리 작업은 시스템 관리자 또는 특별한 권한을 가진 지정된 사용자에게 의해서만 수행된다.

### ■ 루트 사용자

루트는 시스템에서 무제한의 특권을 갖는다. 이것은 다시 말해서 루트인 사용자가 가장 큰 위험을 초래할 수도 있다는 것을 의미한다. 그러므로 일상 업무를 수행할 때는 제한된 액세스를 가진 사용자로서 로그인 하는 것이 좋은 습관이다. 이렇게 하면 오류가 발생되더라도 쉽게 복구할 수 있다. Su (switch user) 명령어로 사용자 ID를 변경할 수 있으며 암호를 알고 있는 어떠한 사용자의 권한도 획득할 수 있다. 루트인 관리자는 이 레벨에서 실수를 하게 되면 큰 재앙을 초래할 수도 있기 때문에 조심해야 한다.

루트로서 수행하는 작업을 제한하는 또 다른 이유는 루트로 바로 로그인하는 경우 WSM과 SMIT 같은 응용프로그램이 루트의 홈 디렉토리, /(루트)에 파일을 생성한다는 것이다. 이 디렉토리는 루트 파일 시스템의 일부이다. 만약 루트 파일 시스템이 채워지면 시스템이 다운된다.

#### ■ su 명령어 사용

루트 사용자로서 su 명령어를 사용하여 다른 사용자로 전환하게 되면, 사용자 고유의 환경은 그대로 남지만 새로운 사용자 ID의 권한을 얻게 된다. 이 사용자 ID에서 종료하면 원래 로그인한 상태로 돌아가며 이전 권한을 다시 얻게 된다. su를 사용하여 다른 계정으로 전환하면, 시작한 계정으로부터 종료해야 한다. 만약 한 계정에서 다른 계정으로 전환하고 또 다시 다른 계정으로 전환하면 두 번 종료해야 한다.

## Unit 2. System Management Tools and Documentation

---

시스템 관리 Tool(SMIT)

SMIT Main Menu

대화메뉴(Dialog Screen)

출력메뉴(Output Screen)

smit 명령어



## 시스템 관리 Tool(SMIT)

- 메인 메뉴 : 시스템을 관리하는데 있어, 관련된 분야로 나누는 계층적인 메뉴
- 하위 메뉴
- 선택자 : 사용자가 값을 입력하는 한 개의 옵션
- 대화 메뉴(Dialog Screen) : 작업수행 방법을 제어함
- 팝업(pop-up) : 확장 가능한 사전 정의 값 들의 목록
- 출력 메뉴(Output Screen)

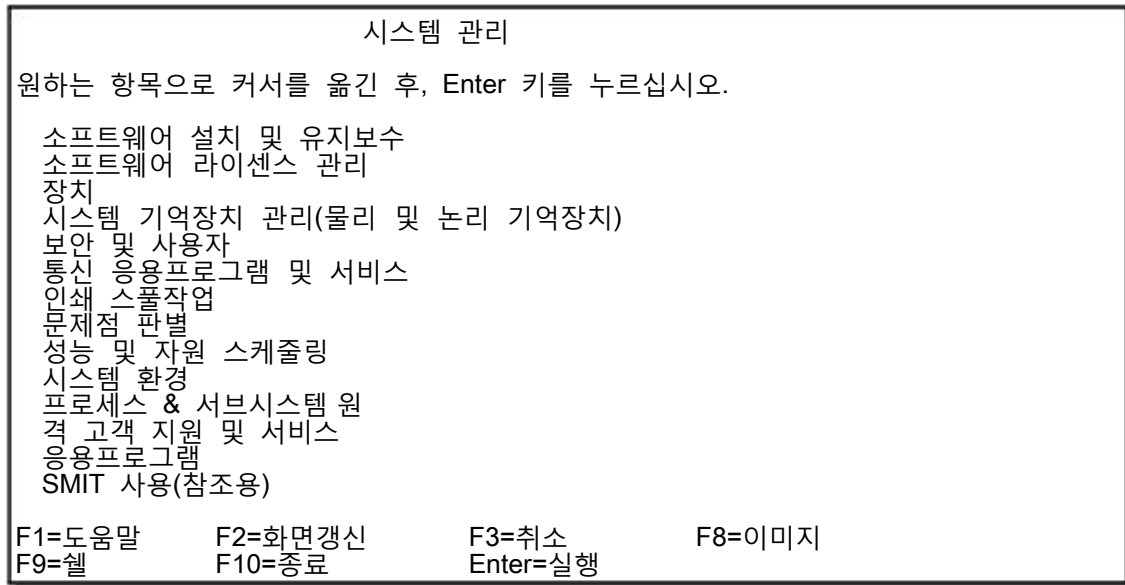
### Notes :

#### ■ SMIT(System Management Interface Tool)

- ▶ 메인 메뉴 : 시스템을 관리할 관련된 분야로 나누는 계층적인 메뉴이다. 사용자는 커서 또는 마우스(모티프 버전의 경우)를 사용하여 메뉴 항목을 선택하여 작업을 시작할 수 있다.
- ▶ 하위 메뉴 : 메뉴와 동일한 방법으로 하위 메뉴를 사용할 수 있다. 사용자는 텍스트를 입력하지 않고서 메뉴 선택을 할 수 있다. 일부 메뉴는 한 군데 이상에서 보여지기도 한다
- ▶ 선택자 : 사용자가 값을 입력하는 한 개의 옵션을 보여준다. 선택메뉴는 대화 메뉴와 유사하지만 명령을 수행하기에 앞서 사용자가 볼 수 있는 마지막 메뉴이다.
- ▶ 대화 메뉴 : 작업 수행 방법을 제어하도록 한다. 예를 들어, 드라이브의 속성을 설정하거나 설치 드라이브를 지정할 수 있다.
- ▶ 팝업 : 팝업 목록의 두 가지 유형은 사전 정의 값들의 고정 집합인 ring과 확장 가능한 사전 정의 값들의 집합인 목록입니다
- ▶ 출력 메뉴 : 이러한 명령의 표준 출력과 표준 오류는 특정 SMIT 출력 메뉴에 보여집니다.

## SMIT Main Menu

### ■ smit(ASCII)



### ■ smit(Motif)



## Notes :

SMIT는 작업에 맞춘 대화를 통해서 새로운 관리자가 시스템 관리를 단순화할 수 있게 한다. SMIT에는 문자 기반 인터페이스 또는 ASCII SMIT와 AIX Windows 그래픽 인터페이스 또는 Motif SMIT의 두 버전이 있다.

## 대화 메뉴(Dialog Screen)

# smit date

날짜 및 시간 변경/표시

입력 필드에 값을 입력하거나 선택하십시오. 원하는 것을 모두 변경한 후, Enter 키를 누르십시오.

	[입력 필드]	
연도 ( 00 - 99 )	[00]	#
월 (01-12)	[Jun]	+
일 (1-31)	[1]	#
* 시 (00-23)	[]	#
* 분 (00-59)	[]	#
초 (00-59)	[]	#

F1=도움말	F2=화면갱신	F3=취소	F4=리스트
F5=재설정	F6=명령	F7=편집	F8=이미지
F9=셸	F10=종료	Enter=실행	

### Notes :

- \* 필수 항목
- # 숫자항목
- / 경로이름이 필요함
- X 16진수 값이 필요함
- ? 값이 표시되지 않음
- + 팝업 목록 또는 링이 사용가능 함

## 출력화면(Output Screen)

명령 상태

명령: OK                      stdout: 예                      stderr: 아니오

명령이 완료되기 전에 아래에 추가적인 지시가 나타날 것입니다.

name	status	location	description
tty0	사용가능	01-S1-00-00	비동기식 터미널
tty1	사용가능	01-S2-00-00	비동기식 터미널
tty2	사용가능	01-S3-00-00	비동기식 터미널

### Notes :

위 내용은 명령을 실행한 후에 피드백 또는 출력을 제공한다. 첫 번째 라인은 실행 성공 또는 실패와 같이 명령에 대한 정보를 보유합니다. 이 경우 명령어 다음의 OK는 오류가 없음을 보여준다. 정보를 제공하는 항목은 세 개로 Command, stdout 및 stderr이다.

Command 항목 : OK, RUNNING 및 FAILED 값을 표시한다.

stdout 항목 : 표준 출력이 생성 되었는지 여부를 나타냅니다. 표시되는 값은 yes와 no입니다. 화면에서 출력패널은 실행된 모든 명령을 보여준다.

stderr 항목 : 오류 메시지가 stderr로 송신되었는지 여부를 보여준다..

## smit 명령어

### ■ smit 명령어 유형

```
# smit[-options] [FastPath]
```

### ■ ASCII버전 호출

```
# smitty
```

### ■ 로그파일과 스크립트 파일의 재지정

```
# smitty -s /home/team01/smit.script
```

## Notes :

SMIT는 두 파일, 즉 smit.script 파일과 smit.log 파일을 생성한다. smit 파일이 기록되도록 요청했는데 해당 디렉토리에 쓰기권한이 없으면 smit.script 파일과 smit.log 파일이 생성되지 않는다. SMIT는 smit.log 파일 또는 smit.script 파일을 겹쳐쓰지 않는다. 가능하다면 파일에 추가된다.

### ■ smit(System Management Interface Tool)

```
# smit [-s | -l ] PathName
```

-s 옵션 : PathName 매개변수가 지정한 파일에 smit.script 파일이 저장.

-l 옵션 : PathName 매개변수가 지정한 파일에 smit.log 파일이 저장.

#### ▶ -s 옵션을 지정하지 않은 경우 :

스크립트 정보는 \$HOME/smit.script 파일에 저장

#### ▶ -l 옵션을 지정되지 않은 경우 :

로그 정보 \$HOME/smit.log 파일에 저장.

## Unit 3. System Startup and Shutdown

---

순서 모드

SYSTEM P Start up Process Overview

alog

etc/inittanb 파일의 내용

시스템 자원관리 명령어

시스템 종료

## 순서 모드

### ■ Normal Mode

로그인 프롬프트를 보여줌  
모든 프로세스가 실행  
다중사용자 모드가 초기화

### ■ System Management Services(CD-ROM, Type 부팅) 기계는 펌웨어로 실행

기계가 부트 목록을 설정

### ■ Maintenance Mode(관리모드)

유지보수 메뉴가 나옴  
루트 사용자의 암호를 복구할 때  
부트 되지 않는 기계를 수리할 때

### ■ Diagnostics

AIX 진단 프로그램 실행

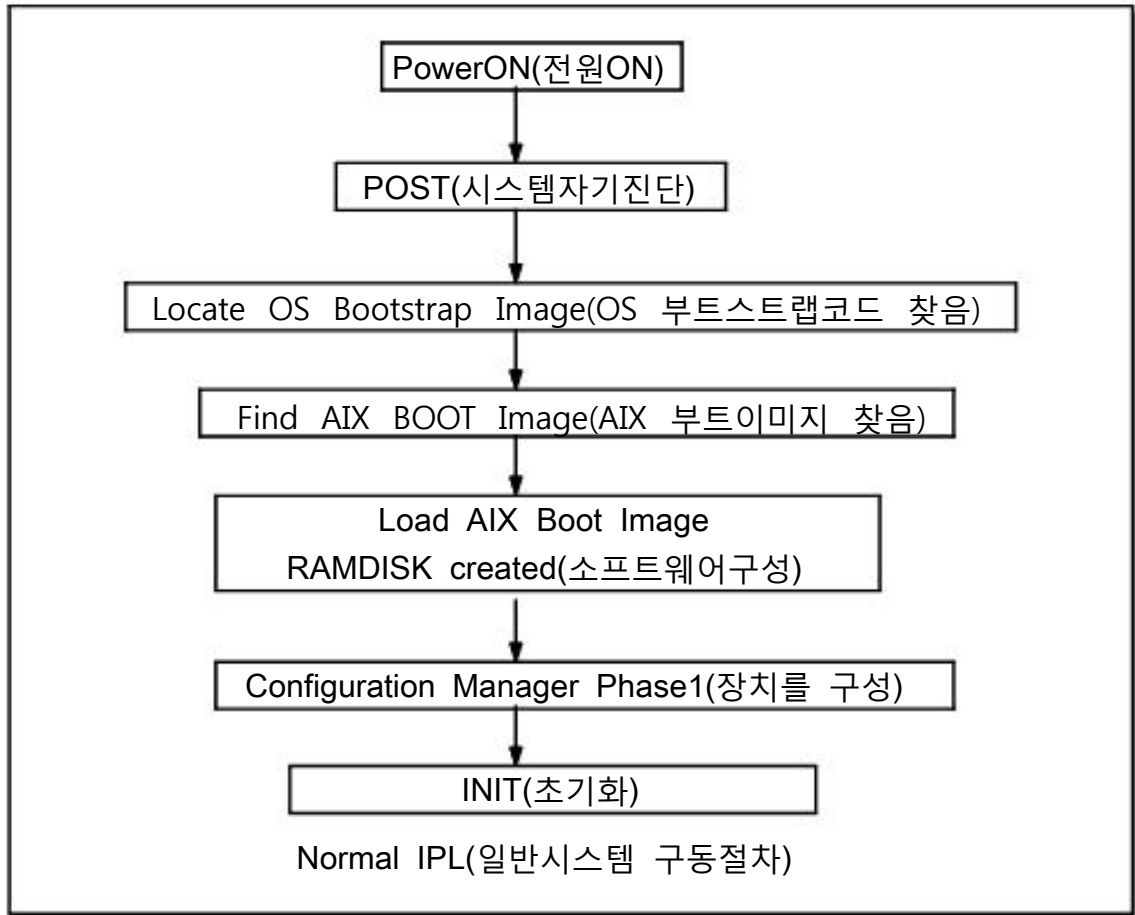
## Notes :

순서모드는 크게 Normal 모드 와 Service 모드로 나눌 수 있다.

### ■ Normal과 Service 모드 비교

	<p>시스템의 전원을 켜서 시작 모든 사용자 및 시스템 관리자가 시스템을 사용한다. Normal 모드 Normal 모드는 다중 사용자 모드라고 불려진다. 모든 단말기와 콘솔을 사용할 수 있다. 모든 시스템 파일을 액세스할 수 있으며, 모든 프로그램 사용 가능.</p>
	<p>CD-ROM 또는 테이프로부터 시동하여 시작한다. 시스템 관리자만이 시스템을 사용할 수 있다. Service 모드는 단일 사용자 모드로도 불려진다. 콘솔 만을 사용할 수 있다. 사용할 수 있는 프로그램과 액세스할 수 있는 파일은 시스템 관리자에 의해 사용될 수 있는 것들 뿐이다.</p>

## SYSTEM P Start up Process Overview



### Notes :

PowerOn : 시동 프로세스는 전원 스위치를 누를 때 시작

POST : PCI RS/6000 시스템의 부트 시, 시스템은 POST(Power On Self-Test)를 수행하는데, 이 과정에서 메모리, 키보드, 통신 어댑터, 그리고 오디오 요소를 초기 한다.

OS Bootstrap Image : PCI RS/6000 시스템은 여러 개의 운영체제를 사용할 수 있기 때문에 시동 단계에서 OS 부트스트랩 코드를 찾는 작업을 한다.

AIX Boot Image : OS 부트스트랩 코드는 AIX 부트 이미지를 찾는다.

Load AIX Boot Image RAMDISK Created (AIX 부트 이미지를 로드) : 시스템은 AIX 부트 이미지를 찾아 이를 로드하고, 부트 이미지의 로드가 끝나면 소프트웨어 구성을 시작한다.



Configuration Manager PHASE 1 : 현재 시스템에 장착된 device를 configure

INIT(초기화)

IPL(initial program loader) : 초기 프로그램 적재기

## alog

- alog란 : 시스템 시작 시 생성되는 모든 메시지를 시스템 관리자가 볼 수 있도록 화면에 나타내는 명령어
- 시스템 시작 중에 alog는 다음과 같은 네 가지 유형의 프로세스로부터 메시지를 수신하여 로그한다.
  - ▷ rc.boot 스크립트 : 파일 시스템의 일관성을 확인(fsck)하고 메모리와 파일 시스템을 동기화하는 프로세스
  - ▷ 구성 관리자 : 장치설정 구성 프로세스
- NIM(Network Install Manager) : 네트워크 구성 프로세스
- 사용자 응용프로그램

## Notes :

### ■ alog 파일의 특성

alog 파일은 순환 파일로, 파일이 채워지면 가장 최근 메시지가 이전 메시지 위에 쓰여지는 것을 의미한다. AIX는 디스크 공간을 절약하기 위해 순환 파일을 사용한다. 사용자는 SMIT를 사용하여 파일 크기를 변경할 수 있다.

### ■ 시스템 시작시 생성되는 모든 메시지를 볼 수 있는 명령어

# alog[-option]

-o 옵션 : LogFile의 내용을 나열합니다. LogFile의 내용을 표준 출력장치에 차례로 기록한다.

-t 옵션 : LogType alog 구성설정 데이터베이스에 정의된 로그를 식별한다.

-f 옵션 : LogFile 로그 파일의 이름을 지정한다.

#### ▶ NIM 로그를 보기

```
# alog -o -t nim
```

#### ▶ 설치과정 로그를 보기

```
# alog -o -t bosinst
```

#### ▶ 현재 날짜와 시간을 mylog이름의 로그파일에 기록

```
# date | alog -f /tmp/mylog
```

## etc/inittab 파일의 내용

```
init:2:initdefault:

brc::sysinit:/sbin/rc.boot 3 >/dev/console 2>&1 # Phase 3 of system boot
load64bit:2:wait:/etc/methods/cfg64 >/dev/console 2>&1 # Enable 64-bit execs
rc:2:wait:/etc/rc 2>&1 | alog -tboot > /dev/console # Multi-User checks

writesrv:2:wait:/usr/bin/startsrc -swritesrv
uprintfd:2:respawn:/usr/sbin/uprintfd

diagd:2:once:/usr/lpp/diagnostics/bin/diagd >/dev/console 2>&1
pmd:2:wait:/usr/bin/pmd > /dev/console 2>&1 # Start PM daemon
logsymp:2:once:/usr/lib/ras/logsymptom # for system dumps
imnss:2:once:/usr/IMNSearch/bin/imnss -start imnhelp >/dev/console 2>&1
httpdlite:2:once:/usr/IMNSearch/httpdlite/httpdlite

wlm:2:once:/usr/sbin/wlmcntrl > /dev/console 2>&1
orakstat:2:wait:/etc/loadext -l /etc/ora_kstat i4ls:2:wait:/etc/i4ls.rc >
/dev/null 2>&1 # Start i4ls
imgss:2:once:/usr/IMNSearch/bin/imq_start >/dev/console 2>&1
l3:3:wait:/etc/rc.d/rc 3
```

### Notes :

/etc/inittab의 모든 파일은 동일한 포맷을 갖고, 식별자가 가장 먼저 나열되며 실행 레벨, 조치, 명령어 순으로 나열된다

- /etc/inittab : /etc/inittab 파일은 시스템에 어떤 프로세스가 초기화되어야 하는지 알려주고 alog로 이동할 메시지를 제어합니다
- 식별자 : 식별자는 프로세스의 이름으로 최대 14자까지 사용할 수 있으며, 단말기는 식별자로 단말기의 논리 디바이스이름을 사용한다.  
식별자(id):runlevel(실행레벨):action(조치):command(명령어)

■ 실행 레벨 : 실행 레벨은 프로세스의 실행 모드를 결정한다. 디폴트 실행 레벨은 2로 다중 사용자 모드이다. 단일 사용자 모드의 실행 레벨은 0이다. S, s, M 그리고 m도 단일 사용자 모드이다. 실행 레벨이 지정되어 있지 않으면 프로세스는 모든 모드에서 실행한다. AIX는 0에서 9까지의 실행 레벨을 사용한다. 실행 레벨을 변경하기 위해 `telinit` 명령어를 사용하면 SIGTERM 신호를 새로운 실행 레벨이 정의되지 않은 모든 프로세스에 보낸다. 20초 후에 프로세스가 종료되지 않으면 SIGKILL 신호를 보낸다. `/etc/inittab` 파일의 항목은 0에서 9까지의 실행 레벨을 가지거나 ::로 표시된다. 이는 프로세스가 모든 레벨로 실행된다는 것을 나타낸다.

■ 조치 : 조치는 `/etc/inittab`이 프로세스가 무슨 일을 해야 하는지를 나타낸다. 다음 목록은 조치에 대한 설명입니다.

- ▶ `spawn` : 프로세스가 시작되어 있지 않으면 프로세스를 시작
- ▶ `once` : 프로세스를 시작 시킵니다. 중단되면 다시 시작
- ▶ `Wait` : 프로세스를 시작 시키고 다음 프로세스를 시작 시키기 전에 완료되길 기다림
- ▶ `Sysinit` : 콘솔을 사용하기 전에 프로세스를 시작시킴

■ 명령어 : 명령어는 동작을 수행

## 시스템 자원 관리 명령어

### ■ List SRC Status(SRC 상태 리스트)

```
# lssrc -g spooler
```

Subsystem	Group	PID	Status
qdaemon	spooler	19624	active
writesrv	spooler	20136	active
lpd	spooler		inoperative

### ■ Start a Subsystem(SRC 명령어가 동작될 서브시스템지정)

```
# startsrc -s lpd
```

0513-059 The lpd Subsystem has been started.Subsystem PID is 12472

### ■ Refresh a Subsystem(서브시스템을 재시작)

```
# refresh -s lpd
```

0513-095 The request for subsystem refresh was completed successfully.

### ■ Stop a Subsystem(서브시스템을 중단)

```
# stopsrc -s lpd
```

0513-044 the lpd Subsystem was requested to stop

## Notes :

- 정의된 그룹, 서브시스템 및 서버서버에 대한 명령어
  - ▶ lssrc [-option] : 정의된 모든 시스템의 목록을 보여줌
  - ▶ startsrc [-option] : 서브시스템을 시작
  - ▶ stopsrc [-option] : 서브시스템을 중단
  - ▶ refresh [-option] : 서브시스템을 재시작 하거나 갱신
  - ▶ trace [-option] : 서브시스템을 추적.(trace에는 on과 off의 두 가지 옵션을 사용)

- s 옵션 : 명령이 지정한 서브시스템에만 동작하도록 지시
- g 옵션 : 명령이 지정한 시스템이 속해 있는 그룹내의 모든 서브시스템에 동작 되도록 지정
- p 옵션 : 지정한 프로세스 ID(PID)에 대해 명령어가 수행
- s 옵션 : 대신 -p 사용할 수 있으면, 이 두 가지 옵션 모두는 명령어가 동작될 서브 시스템을 지정
- a 옵션 : 모든 서브시스템에 대해 명령을 수행하도록 한다.  
lssrc -a 명령은 모든 서브시스템을 나열하게 한다.
- f 옵션 : 강제 중단을 뜻하며 stopsrc 명령에서 사용된다.  
강제 중단을 어플리케이션이 프로세싱을 마칠 때까지 기다리지 않고 즉시 시스템을 종료한다.

## 시스템 종료

- 시스템 종료 : 모든 프로세스, 서비스 및 통신을 포함하여 시스템에 있는 모든 것을 종료함을 의미한다.
- shutdown 명령어를 사용하여 시스템을 종료할 수 있으며 이 명령어는 모든 단말기에 다음의 경고 메시지를 보냅니다.

```
# shutdown +2 The system will be down until 3AM

Broadcast message from root@localhost    (tty) at 1:30:20...

the system will be down until 3AM

shutdown: PLEASE LOG OFF NOW!! All
processes will be killed in 2 minutes
```

### Notes :

- 시스템 종료 명령어

# shutdown [-옵션][+시간 메시지]

-F 옵션 : 빠른 종료를 수행하며 경고 메시지를 보내지 않고, 어플리케이션이 종료 되기를 기다리지 않으며 사용자에게 시간을 주지 않고 종료 스크립트를 수행

-r 옵션 : 시스템 종료 후 리부트하는 옵션으로 경고 메시지를 보내며 정상적으로 종료

-m 옵션 : 유지 보수 모드로 시스템을 종료하는 옵션으로 경고 메시지를 보내며 정상적으로 종료

-k 옵션 : 경고 메시지를 보내지만 시스템은 종료하지 않습니다. 이 옵션은 시스템을 종료해도 좋은지 신속히 결정하기 위해 사용. 경고 메시지에 대해 사용자가 응답을 하지 않으면, 시스템 관리자는 시스템을 종료해 도 안전함

## Unit 4.Devices(장치)

---

Device Terminology

지원되는 모든 장치들의 목록

정의된 모든 장치들의 목록

Device States



## Device Terminology

---

- 물리적 디스크(Physical Devices)
  - 포트(ports)
  - 장치 디스크(Device Drivers)
  - 논리적 디스크(Logical Devices)
  - /dev Directory
- 

### Notes :

- 물리적 디스크 : Hardware를 의미한다.
- Ports : Physical device가 붙는 Physical한 connector나 adapter.
- 장치 디스트 : device로 보내질 data의 format과 port에서의 activity를 control 하는 kernel에 있는 소프트웨어.
- 논리적 디스크 : 물리적 디스크를 user와 application이 access하게 하는 Software interface(special files). 논리적 디스크에 추가된 data는 적당한 device driver로 보내져 읽히게 된다.
- /dev directory : user에 의해 직접 access될 수 있는 논리적 디스크를 포함하는 drectory.

## 지원되는 모든 장치들의 목록

# lsdev -P -H

sys	sys_p	node	시스템 오브젝트
adapter	ssa	pci	IBM SSA 어댑터(14104500)
cpucard	cpucard1	sys	CPU 카드
memory	L2cache	sys	L2 캐쉬
sys	sys1	node	시스템 오브젝트
ipsec	4	IPSEC	IP 버전 4 보안 확장

# lsdev -Pc tape

```
tape 1200mb-c scsi 1.2GB 1/4인치 테이프 드라이브
tape 150mb     scsi 150MB 1/4인치 테이프 드라이브
tape 3490e     scsi 3490E 자동로딩 테이프 드라이브
tape 4mm2gb    scsi 2.0GB 4mm 테이프 드라이브
tape 525mb     scsi 525MB 1/4인치 테이프 드라이브
tape 8mm       scsi 2.3GB 8mm 테이프 드라이브
tape 8mm5gb    scsi 5.0GB 8mm 테이프 드라이브
```

## Notes :

### ■ 시스템 내의 장치의 특성을 표시하는 명령어

# lsdev [-옵션]

-C 옵션 : 조정된 장치 오브젝트 클래스 내의 장치에 관한 정보를 나열한다.(-P와 같이사용 못함)

-P 옵션 : 사전 정의된 장치 오브젝트 클래스 내의 장치에 관한 정보를 나열한다.(-C, -I, -S와 함께 사용할 수 없다.)

## 정의된 모든 장치들의 목록

# lsdev -C -H

sys0	사용가능	00-00	시스템 오브젝트
sysplanar0	사용가능	00-00	시스템 플래너
pci0	사용가능	00-fef00000	PCI 버스
sa2	사용가능	01-S3	표준 I/O 직렬 포트
siokma0	사용가능	01-K1	키보드/마우스 어댑터
ses0	사용가능	10-60-00-15,0	SCSI 격납장치 서비스 장치
lvdd	사용가능		LVM 장치 드라이버

# lsattr -EH -l sys0

attribute	value	description	user_settable
keylock	normal	부트시 시스템 키잠금 상태	거짓
maxbuf	20	블록 I/O 버퍼 캐쉬에 있는 최대 페이지 수	참

# lsattr -E -l sys0 -a realmem

realmem 131072 사용 가능한 물리적 메모리의 크기(KB) 거짓

## Notes :

- 한 장치의 모든 속성이나 지정된 특정 속성에 대한 현재 또는 생략시 속성값 일람 주어진 장치의 속성 또는 장치의 종류에 관한 정보를 표시하는 명령어

# lsattr

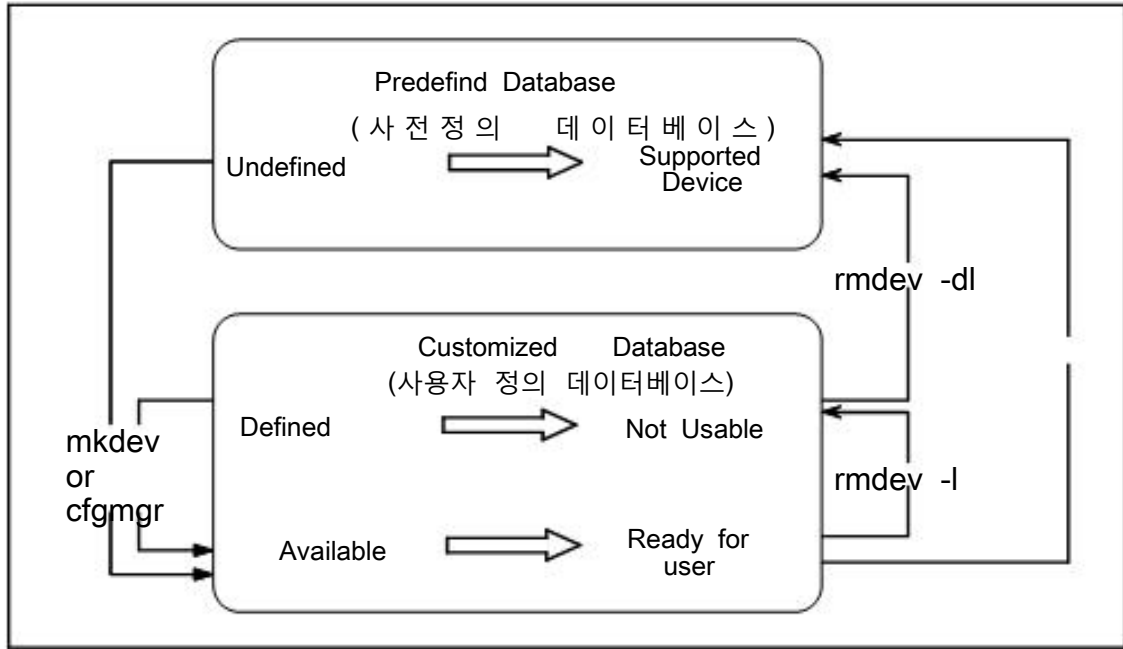
- ▶ lsattr -E -l et0: 이더넷0번에 대한 장치구성 정보를 보여준다.

- 시스템에 대한 구성설정, 진단 및 주요 제품 데이터 정보를 표시하는 명령어

# lscfg

- lsdev -C -H 옵션은 Resource name에서 현재의 상태와 Address또는 location 그리고 Customized Database안에 모든 디바이스의 정보를 보여준다(헤더 포함)

## Device States



### Notes :

시스템에 알려지지 않은 장치와 장치의 특성, 정보를 시스템이 가지고 있으나 운용이 불가능한 상태의 장치를 `# mkdev` or `# cfgmgr` 명령어를 이용하여 새롭게 정의하고 사용 가능하게 한다. `# rmdev` 명령어를 이용하여 장치를 삭제하고 해체한다.

#### ■ 사전 정의 데이터베이스(Predefined Database) :

사전 정의 데이터베이스는 AIX가 지원하는 모든 장치 목록을 포함한다.

#### ■ 사용자 정의 데이터베이스(Customized Database) :

사용자 정의 데이터베이스는 시스템에 구성되고 실제 시스템에 있는 장치의 목록을 포함한다.

#### ■ Undefined : 시스템에 알려지지 않은 상태

#### ■ Defined : 장치의 특성, 정보를 시스템이 가지고 있으나 운용이 불가능한 상태

#### ■ Available : 장치가 사용 가능한 상태

#### ■ lsattr : 한 장치의 모든 속성이나 지정된 특정 속성에 대한 현재 또는 생략 시 속성값을 보여준다

#### ■ chdev : Available 또는 Defined 상태의 device 속성변경

#### ■ cfgmgr : 시스템이 작동 한 후 추가된 장치에 대한 self configuration

#### ■ rmdev -dl : 장치 구성의 삭제

#### ■ rmdev -l : 장치구성의 해체

## Unit 5. Logical Volume Manager

---

LVM이란?

전통적인 UNIX 저장장치

LVM의 이점

Volume Group

볼륨그룹 지시자 영역(VGDA)

Volume Group의 제한

Volume Group의 추가/삭제

Volume Group의 특성 설정

Volume Group의 변경

Volume Group의 활성화/비활성화

Volume Group의 반출입

Volume Group의 재생성

Physical Volume

SMIT Physical Volumes 메뉴

Physical Volume의 정보 리스트

Physical Volume의 추가/이동

Logical Volume

Logical Volume의 사용

Logical Volume의 정보 리스트

Mirroring

Striping

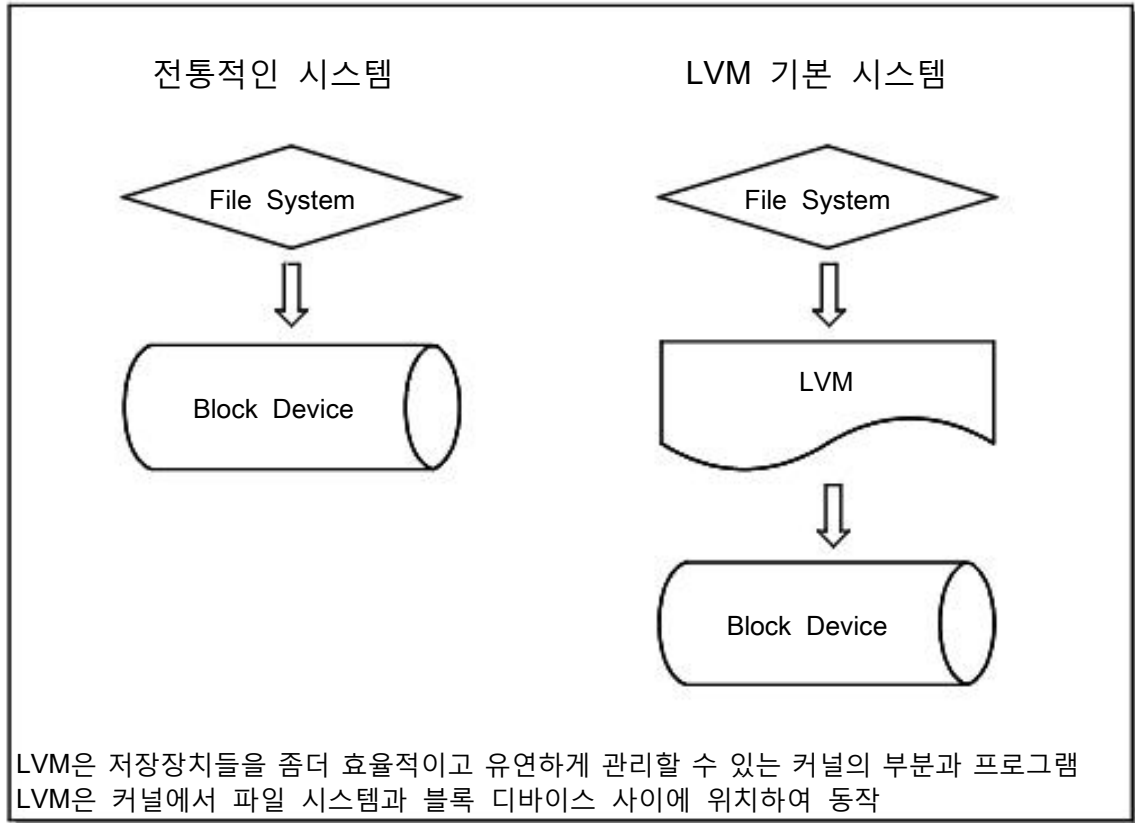
Logical Volume의 정책

Logical Volume의 특성 보기

Logical Volume의 추가

Logical Volume의 삭제

## LVM이란?



### Notes :

LVM이라는 볼륨관리 시스템이 있음으로써 디스크 관리에 탁월한 장점이 있다.  
그림의 '전통적인 시스템'과 'LVM 기본 시스템'의 차이점은 다음의 '전통적인 Unix 저장장치와 LVM의 이점'에서 알 수 있음.

LVM(Logical Volume Manager) :

저장장치들을 좀더 효율적이고 유연하게 관리할 수 있는 커널의 부분과 프로그램을 말한다.

여러 물리적인 디스크들을 하나의 논리적인 디스크처럼 다룰 수 있게 함으로서 조합 방법에 따라 고용량, 고속, 데이터의 무결성을 실현하는 점이다.

## 전통적인 UNIX 저장장치



### Notes :

- 전통적인 UNIX 디스크 저장장치 관리는 시스템 관리자에게는 매우 어려운 일이었다.
- 시스템에 디스크 추가할 때, 관리자는 디스크를 파티션으로 나눠야 했다. 파티션은 관련된 파일과 디렉토리가 포함된 파일 시스템을 만들기 위한 것이었다.
- 파티션의 사이즈를 추가하기가 어려웠다.  
 사용자는 시스템을 종료하고, 크기가 변경되는 파티션이나 이 파티션 변경의 영향을 받는 모든 파티션의 데이터를 백업하며, 디스크를 다시 파티셔닝하고, 데이터를 복구해야만 시스템을 다시 사용할 수 있었다.  
 이것은 매우 어려웠기에 시스템이 한가한 밤에 주로 이루어졌다. 만약 피크타임에 시스템이 부족하게 되면, 관리자는 동적으로 공간을 추가 할당할 수가 없었다. 따라서 사용자가 시스템에서 빠져 나와야 했다
- 연속된 공간이 필요했기 때문에, 여러 개의 디스크에 걸쳐서 만들 수가 없었다. 파일 시스템의 크기에 한계가 있었으며, 큰 파일 시스템의 경우 가장 큰 디스크에 할당해야만 했다.

## LVM의 이점

- Logical Volume은 비연속적인 공간이다.
- Logical Volume은 여러개의 disk를 걸칠 수 있다.
- Logical Volume size 를 쉽게 변환할 수 있다.
- Mirroring이 가능하다.
- 시스템에 hard disk를 쉽게 붙일 수 있다.

### Notes :

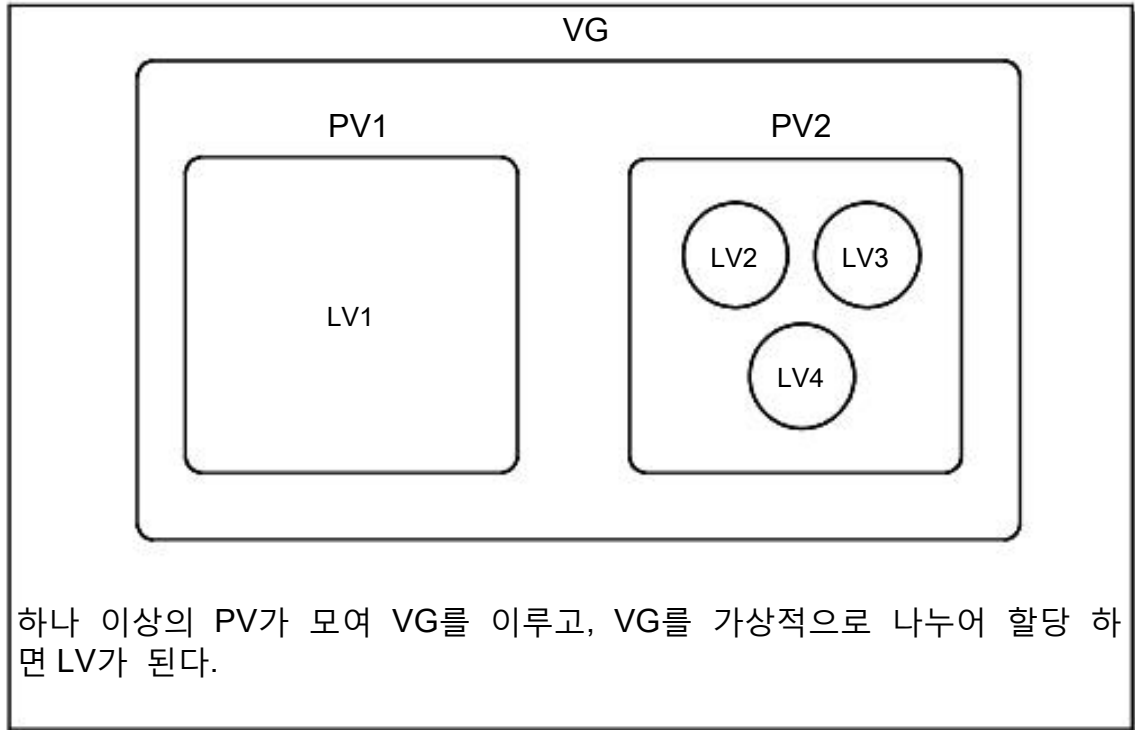
LVM(Logical Volume Manager)은 디스크 공간을 논리적 레벨에서 관리한다. 논리적 공간과 물리적 공간을 연결함으로써 비연속적인 공간에 데이터를 저장하고, 다수의 디스크에 걸쳐서, 복제하고, 동적으로 공간을 늘리며 하드 디스크를 관리한다.

PC급 시스템 사용자들의 가장 큰 고민 중의 하나가 '시스템을 처음 설치할 때에 파티션의 구성을 어떻게 할 것인가' 이다. 파티션은 한번 구성해 놓으면 바꾸기가 쉽지 않기 때문이다. 또 파티션이 가득 차기라도 하면 어렵게 백업을 하거나 자료를 지워야 할 경우도 심심치 않게 생긴다. 하지만 LVM을 쓰면 간단하게 저장공간을 확장, 축소 할 수 있기 때문에 그런 고민을 덜 수 있다.

중소형 서버에서는 비교적 적은 비용으로 대용량 저장 장치를 구현하는 것이 가능해진다. 그리고, 백업 없이 기존의 환경을 유지한 채 확장이 가능하기 때문에 유지보수면에서 상당한 이득이 있다.



## Volume Group



### Notes :

Volume Group(VG) :

시스템 내에서 15자 이내의 유일한 이름을 가진다.

다양한 크기와 유형을 갖는 1-32개의 physical Volume으로 구성된다.

AIX 시스템은 1-225개의 Volume group을 갖는다.

변경, 확장, 축소, 생성, 열람, 반출(제거), 반입(설치), 재구성 할 수 있다.

시스템 설치 후, 시스템은 시스템을 시동하는데 필요한 logical Volume의 기준 집합을 구성하는 하나의 Volume과 설치각본에서 지정하는 다른 Volume group으로 구성된다. 또한 Volume group이나 이미 생성된 다른 Volume group에 추가 할 수 있다. 소형 시스템은 시스템에 부착된 모든 Physical Volume을 포함하기 위하여 하나의 Volume group이면 충분하지만, 대형 시스템은 추가적인 Volume group이 필요하다.

시스템을 설치할 때, 첫 볼륨 그룹이 생성된다. 이것이 바로 rootvg이다. 사용자의 rootvg은 시스템을 시작할 때 필요한 논리적 볼륨들과 설치하는 스크립트에서 결정된 다른 논리적 볼륨들로 이루어진다. rootvg은 페이징 스페이스, 저널 로그, 부트 데이터, 덤프 공간 각각을 별도의 논리적 볼륨에 포함하고 있다.

rootvg는 내장 디스크만으로 구성하기를 권장한다.

만약 새로운 PV가 추가될 때에는 rootvg에 추가하거나 PV를 이용해 새로운 VG를 생성 하여 추가할 수 있다. 시스템당 255 볼륨 그룹까지 포함될 수 있다.

하드 디스크를 추가하는 것은, RS/6000 박스 내에 뿐만 아니라, 외장으로도 추가될 수 있다.

이런 외장 디스크들은 누구나 접근할 수 있으므로, 안전하지 않을 수 있다. 그러므로 rootvg는 외장 디스크에 저장하지 않는 것이 좋다.

개별적인 볼륨 그룹에 운영체제를 저장하는 것이 좋다.

만약 데이터가 개별적으로 저장되면, 운영체제가 업그레이드되는 동안, 데이터를 보호 하기 위해, 임시적으로 제거되거나, 오프라인 시킬 수 있다. 보안을 위해 쉽게 데이터를 제거할 수 있고 볼륨 그룹 레벨에서 데이터를 관리할 수 있다.

볼륨 그룹별로 백업할 수도 있다. 따라서 VG가 32PV를 포함할 수는 있지만 4PV 이하를 유지하는 것이 좋다.

4개의 디스크에 백업하면 시간이 덜 소요되며, 백업을 통해 복구해야 할 경우 시간을 절약할 수 있다. 4개가 넘는 디스크를 사용하면, 디스크 관리가 어려워지고 성능이 저하 될 수 있다.

## 볼륨그룹 지시자 영역(VGDA)

- ▣ PV당 적어도 1개 이상 가지고 있어야 한다.
- ▣ VG전체에 대한 정보 를 담고 있다.  
(VG의 이름, 상태, 속해 있는 물리적볼륨(PV), 논리적볼륨(LV)들, 물리적파티션(PP), 논리적파티션(LP)들의 할당상태 등)
- ▣ 일반적으로 PV당 하나의 VGDA 할당.
- ▣ 같은 VG에 속해 있는 PV들은 VGDA의 내용이 같다.  
# varyonvg : VGDA의 quorum을 사용가능하게 활성화 시킴.

### Notes :

볼륨 그룹 지시자 영역(VGDA-Volume Group Descriptor Area)은 전체 논리 볼륨의 리스트, 전체 물리적 볼륨의 리스트 같은 전체 볼륨 그룹의 정보를 하드 디스크에 저장하고 있는 영역이다.

일반적으로 물리적 볼륨(PV)당 하나의 VGDA가 존재한다. 단, 볼륨 그룹(VG)에 하나 또는 두개의 물리적 볼륨(PV)이 있을 때에는 예외이다.

각 볼륨 그룹들은 VG를 설명하는 하나의 VGDA 정보를 갖게 된다.

Non-rootvg 볼륨 그룹은 varying on과 varying off 라고 불리는 프로세스를 통해 오프라인으로 또는 온라인으로 접근 가능하다. 시스템은 특정 볼륨 그룹에 대한 VGDA의 가용성을 검사하여, 볼륨 그룹의 varying on과 varying off를 결정한다.

## Volume Group의 제한

### ■ Normal Volume Groups(mkvg)

물리적 디스크	나눌 수 있는 최대의 파티션
1	1016
2	2032
4	4064
8	8128
16	16256
32	32512

### ■ Big Volume Groups(mkvg -B)

물리적 디스크	나눌 수 있는 최대의 파티션
1	1016
2	2032
4	4064
8	8128
16	16256
32	32512
64	65024
128	130048

## Notes :

### ■ Normal Volume Groups :

SMIT나 mkvg 명령어로 Volume Groups이 만들어질 때 Normal Volume Groups 이 기본으로 정해져 있다. 하나의 볼륨에 최대 32개의 디스크를 추가할 수 있고, 최대 32,512개의 파티션을 나눌 수 있다.

### ■ Big Volume Groups :

mkvg -B 명령어로 만들어지며, SMIT는 지원하지 않는다.

하나의 볼륨에 최대 128개의 디스크를 추가할 수 있고, 최대 130,048개의 파티션을 나눌 수 있다.

## Volume Group의 추가/삭제

# smit mkvg

볼륨 그룹 추가			
입력 필드에 값을 입력하거나 선택하십시오. 원하는 것을 모두 변경한 후, Enter 키를 누르십시오.			
	[입력 필드]		
볼륨 그룹 이름	<input type="text"/>		
물리적 공간 크기(MB)	4		+
* 물리적 볼륨 이름	<input type="text"/>		+
시스템 재시작시 볼륨 그룹을 자동으로 활성화합니까?	예		+
볼륨 그룹 MAJOR NUMBER	<input type="text"/>		+#
VG 동시 기능을 생성합니까?	아니오		+
동시 모드로 Auto-varyon합니까?	아니오		+
F1=도움말	F2=화면갱신	F3=취소	F4=리스트
F5=재설정	F6=명령	F7=편집	F8=이미지
F9=헬프	F10=종료	Enter=실행	

# smit reducevg : 볼륨그룹 삭제시 명령어 모드

### Notes :

#### ■ 볼륨그룹 생성시 사용하는 명령어

# mkvg [-옵션] PhysicalVolume

-B 옵션 : Big VG을 생성할 수 있다. Big VG에는 128개의 PV와 512개의 LV를 포함할 수 있다. 이 옵션 없이 VG를 생성할 경우에는, 최대한 32개의 PV와 512개의 LV를 포함할 수 있다.

-f 옵션 : 이미 다른 VG에 포함된 PV가 아니라면, 그 PV를 지정하여 VG를 할 수 있다.

-y 옵션 : VG의 이름을 지정할 수 있다.

-n 옵션 : 시스템이 시작될때 자동으로 볼륨그룹을 활성화 시키지 않는다.

#### ▶ 볼륨그룹생성하기

# mkvg -f -y datavg -s 16 -B hdisk2

■ 볼륨그룹 삭제시 사용하는 명령어

```
# reducevg -df VolumeGroup PhysicalVolume
```

※ 참고

▶ rootvg를 mirroring 하는 절차

(rootvg의 물리적 디스크는 hdisk0 이고, 다른 디스크는 hdisk1로 명시)

1. # extendvg -f rootvg hdisk1 (hdisk1 활성화)
2. # chvg -Qn rootvg(Quorum을 NO 설정)
3. # mirrorvg -S rootvg hdisk0 hdisk1(자동 Sync)
4. # bosboot -ad /dev/hdisk1(부트이미지생성)
5. # bootlist -m normal hdisk0 hdisk1(부트이미지추가)

▶ rootvg mirror 제거

1. # unmirrorvg rootvg hdisk1
2. # reducevg -df rootvg hdisk1
3. # chpv -C rootvg hdisk1
4. # bosboot -ad /dev/hdisk0
5. # bootlist -m normal hdisk0
6. # shutdown -Fr

## Volume Group의 특성 설정

# smit vgsc

볼륨 그룹의 특성 설정

원하는 항목으로 커서를 옮긴 후, Enter 키를 누르십시오.

볼륨 그룹 변경  
 볼륨 그룹에 물리적 볼륨 추가 볼륨  
 그룹으로부터 물리적 볼륨 제거 볼륨  
 그룹 재구성

F1=도움말	F2=화면갱신	F3=취소	F8=이미지
F9=셸	F10=종료	Enter=실행	

### Notes :

#### ■ 볼륨 그룹에 물리적 볼륨 추가

PV를 추가하려면, SMIT에서 "볼륨 그룹에 물리적 볼륨 추가"을 선택하십시오. PV는 시스템에 내장되어 있거나, 외장에 연결되어야 하며, 전원이 켜져 있어야 한다. extendvg 명령어로 이런 작업을 할 수 있다.

- ▶ 명령행에서 datavg에 hdisk2를 추가하기  
 # extendvg -f datavg hdisk2

#### ■ 볼륨 그룹에서 물리적 볼륨 삭제

VG를 삭제하기 위해서 "볼륨 그룹으로부터 물리적 볼륨 제거" 메뉴 혹은 reducevg 명령어를 이용할 수 있다. VG에서 디스크가 삭제된 후에는 그 디스크의 데이터에 접근할 수 없으므로, 그전에 디스크의 데이터를 백업하거나 데이터를 다른 디스크로 옮겨야 한다.

# reducevg [-d] 볼륨그룹명 hdiskn

-d 옵션 : 논리적 볼륨을 지정된 물리적 볼륨(PV)에서 삭제

## Volume Group의 변경

# smit chvg

볼륨 그룹 변경			
입력 필드에 값을 입력하거나 선택하십시오. 원하는 것을 모두 변경한 후, Enter 키를 누르십시오.			
		[입력 필드]	
* 볼륨 그룹 이름	rootvg		
* 시스템 재시작시 볼륨 그룹을 자동으로 활성화합니까?	예		+
* 볼륨 그룹을 온라인으로 유지하기 위해 디스크 쿼럼이 필요합니까?	예		+
이 VG를 동시 가능으로 변환합니까?	아니오		+
* VG를 동시 모드로 Autovaryon합니까?	아니오		+
F1=도움말	F2=화면갱신	F3=취소	F4=리스트
F5=재설정	F6=명령	F7=편집	F8=이미지
F9=셸	F10=종료	Enter=실행	

### Notes :

#### ■ 볼륨그룹 활성화

chvg 명령어는 시스템 시작 중에 볼륨 그룹이 자동으로 활성화되는지 여부를 지정한다. 자주 사용되지 않는 볼륨 그룹은 시스템 시작시 활성화되지 않도록 하는 것이 커널 자원(메모리)의 절약을 위해 바람직할 것이다.

▶ 볼륨그룹 시동 시 자동으로 활성화하기  
# chvg -a y VolumeGroup

▶ 볼륨그룹 시동 시 자동으로 활성화 되지 않게 하기  
# chvg -a n VolumeGroup



## Volume Group의 활성화/비활성화

# smit varyonvg/varyoffvg

볼륨 그룹 활성화			
입력 필드에 값을 입력하거나 선택하십시오. 원하는 것을 모두 변경한 후, Enter 키를 누르십시오.			
	[입력 필드]		
* 볼륨 그룹 이름			+
실효(stale) 물리적 공간을 재동기화합니까?	예		+
볼륨 그룹을 시스템 유지보수 모드에서 활성화합니까?	아니오		+
볼륨 그룹을 강제로 활성화합니까?	아니오		+
경고--이렇게 하면 자료 무결성이 상실될 수도 있습니다.			
동시 모드에서 VG를 Varyon합니까?	아니오		+
F1=도움말	F2=화면갱신	F3=취소	F4=리스트
F5=재설정	F6=명령	F7=편집	F8=이미지
F9=셸	F10=종료	Enter=실행	

### Notes :

#### ■ Volume Group의 활성화

# varyonvg [-옵션] VolumeGroup

-c 옵션 : 동시 모드에서 볼륨 그룹을 연결 변환

-f 옵션 : 사용 가능한 디스크 쿼럼이 없는 볼륨 그룹을 활성 상태로 만든다. -n

옵션 : 볼륨그룹 내에서 효력이 상실된 물리적 공간의 동기화를 불가능하게 함

#### ▶ 볼륨 그룹 vg03을 활성화하기

# varyonvg vg03

#### ■ Volume Group의 비활성화

# varyoffvg [-옵션] VolumeGroup

-s 옵션 : 논리적 볼륨 명령만 볼륨 그룹에서 사용할 수 있도록 볼륨 그룹을 시스템관리 모드 함

#### ▶ 볼륨 그룹 vg03을 비활성화하기

# varyoffvg vg03

## Volume Group의 반출입

# smit importvg

볼륨 그룹 반입			
입력 필드에 값을 입력하거나 선택하십시오. 원하는 것을 모두 변경한 후, Enter 키를 누르십시오.			
		[입력 필드]	
볼륨 그룹 이름			
* 물리적 볼륨 이름			+
볼륨 그룹 MAJOR NUMBER			+#
이 VG를 동시 가능으로 합니까?	no		+
동시VG를 디폴트 연결 변환(vary on)합니까?	no		+
F1=도움말	F2=화면갱신	F3=취소	F4=리스트
F5=재설정	F6=명령	F7=편집	F8=이미지
F9=셸	F10=종료	Enter=실행	

# smit exportvg : 볼륨 그룹의 정의를 반출

### Notes :

- 물리적 볼륨 세트에서 볼륨 그룹의 정의를 반출  
# exportvg VolumeGroup

- ▶ 볼륨 그룹 vg02를 시스템에서 제거하려면  
# exportvg vg02

- 물리적 볼륨(PV)들의 집합에서 새 볼륨 그룹 정의를 반입  
# importvg [-옵션] VolumeGroup PhysicalVolume  
-y 옵션 : 신규 볼륨 그룹에 사용할 이름을 지정

- ▶ 물리적 볼륨 hdisk07로부터 볼륨 그룹 bkgv를 반입하려면  
# importvg -y bkgv hdisk07

### ※ 참고

명령어나 SMIT을 이용하여 다음의 절차대로 VG를 export 할 수 있다.

1. VG의 파일 시스템을 unmount
2. VG의 비활성화
3. VG의 export

## Volume Group의 재생성

### ■ Re-creates a Volume Group

```
# recreatevg [-y VName ] [-p ] [-f ] [-Y Lv_Prefix | -l LvNameFile ] [-L
Label_Prefix ] [-n ] PName
```

```
# recreatevg -y myvg -Y mylv -L myfs hdisk2
```

### Notes :

#### ■ recreatevg 명령어 :

디스크 대 디스크로 복사할 때 사용하고 이것은 사본을 만드는 게 아니라, 독립적인 Volume을 생성한다.

직접적인 디스크 대 디스크의 복사는(dd명령어와 같은) VGDA, LV들을 포함한 디스크의 모든 정보를 다른 디스크로 복사하면서 문제를 발생시킬 수 있다.

recreatevg 명령어를 이용하면, VG, LV, 파일 시스템의 마운트 포인트의 중복을 걱정하지 않아도 된다.

옵션을 이용하여 LV의 이름을 지정할 수 있다.

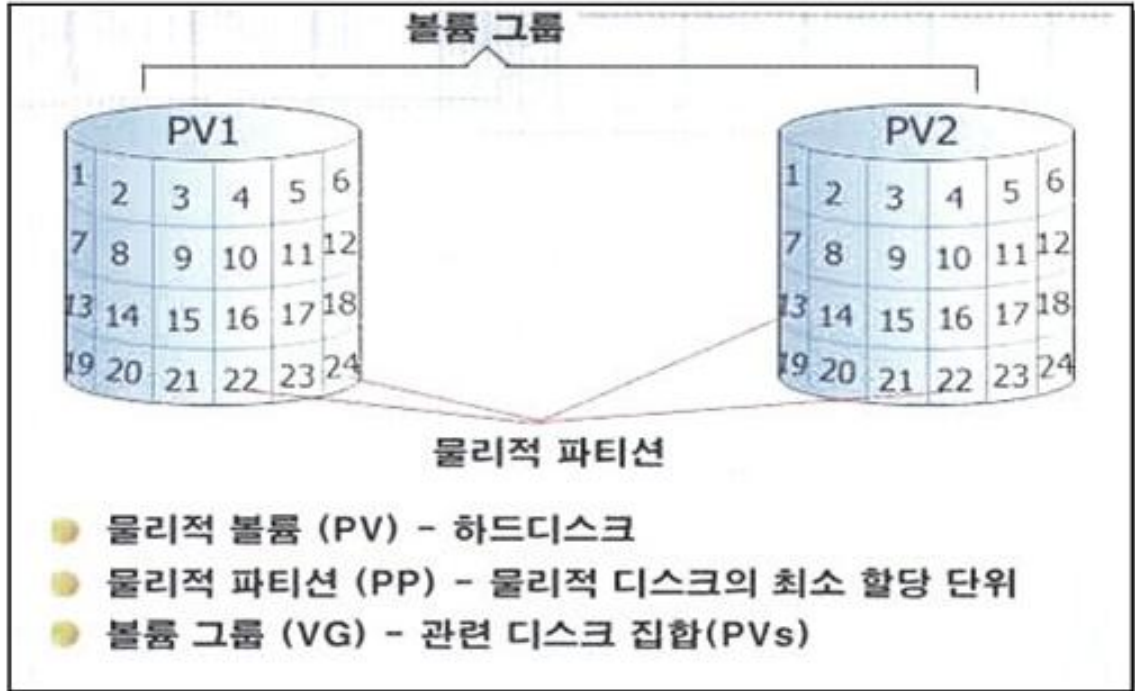
# recreatevg [-옵션] PName

- f 옵션 : 디스크를 모두 가지고 있는 얇은 볼륨그룹을 다시 작성
- l 옵션 : 논리적볼륨 이름을 NameFile에서 지정한 이름으로 변경
- n 옵션 : recreatevg 후에 볼륨그룹이 반입되지만 연결 해제되도록 지정
- y 옵션 : 볼륨그룹 이름을 지정
- Y 옵션 : 다시 작성될 볼륨그룹에서 논리적 볼륨 이름의 접두어가 이 접두어로 변경

#### ▶ hdisk0의 볼륨그룹을 새 이름 testvg로 다시 작성하기

```
# recreatevg -y testvg hdisk0
```

## 물리적 볼륨(Physical Volume)



### Notes :

Physical Volume(pv) : 실질적인 디스크나 하드디스크의 이름

pv를 시스템을 추가할 때 hdisk#으로 나간다. VG(Volume Group)은 저장장치의 단위 중 제일 크다. LVM은 PP를 디스크 할당의 가장 작은 단위로 취급한다.

PP는 LV와 같은 사이즈이거나 동적으로 변화되어질 수 없고 기본 사이즈는 4MB이다. VG는 최소한 하나의 PV에서 32PV까지를 포함하고 하나의 PV는 오직 하나의 VG에만 속할 수 있다.

PP는 하나의 VG에서는 동일한 크기여야 한다.

그러나 시스템에 여러 개의 VG가 있다면, 각각의 VG는 서로 다른 PP 크기를 가질 수 있다. PP는 (1-1024MB)이며 AIX 버전 4.3.2 이상부터는 PV당 가질 수 있는 PP의 개수를 1016개 이상 지원한다. PV당 PP의 개수는 1016의 배수로 지원된다. PV당 1016개 이상의 PP를 사용하고자 하면, VG가 포함할 수 있는 PV의 전체 개수는 줄어든다.

## SMIT Physical Volume 메뉴

# smit pv

물리적 볼륨들			
원하는 항목으로 커서를 옮긴 후, Enter 키를 누르십시오.			
시스템 내의 모든 물리적 볼륨 나열 디스크 추가 물리적 볼륨의 특성 변경 물리적 볼륨의 내용 나열 물리적 볼륨 내용 이동			
F1=도움말	F2=화면갱신	F3=취소	F8=이미지
F9=첼	F10=종료	Enter=실행	

### Notes :

■ 볼륨 그룹내의 물리적 볼륨(PV)에 관한 정보를 표시하는 명령어

# lspv [-옵션] PhysicalVolume

- L 옵션 : 볼륨 그룹에서의 잠금을 확보하기 위한 대기를 지정
- l 옵션 : 물리적 볼륨(PV)의 각 논리적 볼륨(LV)에 대한 다음 필드를 나열
- M 옵션 : 물리적 볼륨(PV)의 각 논리적 볼륨(LV)에 대한 다음 필드를 나열
- p 옵션 : 물리적 볼륨(PV)에서 각 물리적 공간에 대한 다음 필드를 나열

## Physical Volume의 정보 리스트

# lspv : 볼륨 그룹내의 물리적 볼륨(PV)에 관한 정보를 표시

```
hdisk0      000c28cdfc447c03  rootvg
hdisk1      000c28cd53d22ff7  rootvg
```

# lspv hdisk0 : hdisk0 디스크에 대한 정보 표시

```
PHYSICAL VOLUME:hdisk0          VOLUME GROUP: rootvg
PV IDENTIFIER:    000c28cdfc447c03 VG IDENTIFIER  000c28cd53d234c7
PV STATE:        active
STALE PARTITIONS:0              ALLOCATABLE:   yes
PP SIZE:         32 megabyte(s)  LOGICAL VOLUMES:
TOTAL PPs:       271 (8672 megabytes)  VG DESCRIPTORS:
FREE PPs:        71 (2272 megabytes)
USED PPs:        200 (6400 megabytes)
FREE DISTRIBUTION3..00..00..00..38
USED DISTRIBUTION2..54..54..54..16
```

# lspv -l hdisk0 : 물리적 볼륨(PV)의 각 논리적 볼륨(LV)에 대한 필드를 나열

```
hdisk0:
LV NAME          LPs  PPs  DISTRIBUTION  MOUNT POINT
hd5              1    1    01..00..00..00..00  N/A
hd2              92   92    17..00..41..34..00  /usr
hd1              2    2    01..00..00..01..00  /home
hd3              14   14    02..00..00..12..00  /tmp
hd9var           8    8    01..00..00..07..00  /var
hd6              64   64    00..54..10..00..00  N/A
hd4              2    2    00..00..02..00..00  /
hd7             16   16    00..00..00..00..16  N/Aa
```

# lspv -p hdisk0 : 물리적 볼륨(PV)에서 각 물리적 공간에 대한 필드를 나열

```
hdisk0:
PP RANGE  STATE  REGION  LV NAME          TYPE  MOUNT POINT
1-1       used   outer edge  hd5              boot  N/A
2-34      free   outer edge
35-51     used   outer edge  hd2              jfs   /usr
52-52     used   outer edge  hd1              jfs   /home
53-54     used   outer edge  hd3              jfs   /tmp
55-55     used   outer edge  hd9var           jfs   /var
56-109    used   outer middle hd6              paging N/A
121-122   used   center      hd4              jfs   /
164-192   used   inner middle hd2              jfs   /usr
193-198   used   inner middle hd3              jfs   /tmp
207-210   used   inner middle hd2              jfs   /usr
```

## Physical Volume 추가/이동

### ■ Add a disk:

```
# cfgmgr
```

장치를 구성설정하고 선택적으로 장치 소프트웨어를 시스템으로 설치

### ■ Move the contents of a Physical Volume:

```
# migratepv -l lv02 hdisk0 hdisk6
```

한 물리적 볼륨에서 하나 이상의 다른 물리적 볼륨으로 이동

## Notes :

### ■ cfgmgr 명령어 :

장치의 구성을 설정하고 선택적으로 장치 소프트웨어를 시스템으로 설치한다.  
장치구성설치 데이터베이스의 부분이 수행되면서 디스크를 인식하고 구성한다.

#### ▶ Scsi0 어댑터에 접속된 장치 구성설정하기

```
# cfgmgr -l scsi0
```

-l 옵션 : 명명된 장치가 해당 부속 장치와 함께 구성 되도록 지정

### ■ migratepv 명령어 :

할당된 물리적 공간을 한 물리적 볼륨에서 하나 이상의 다른 물리적 볼륨으로 이동 또는 디스크 장애가 발생하거나, 한 디스크에 지나치게 많은 데이터가 할당된 경우에 이용 한다.

#### ▶ hdisk1에서 hdisk6과 hdisk7로 물리적 공간을 이동시키기

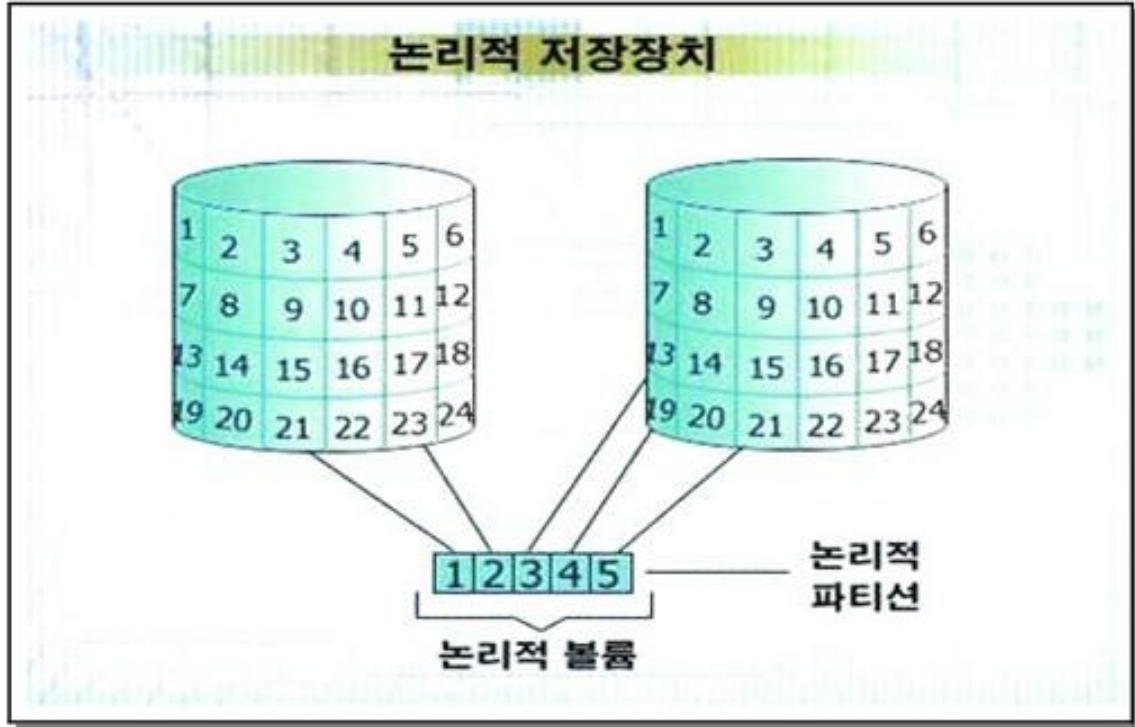
```
# migratepv hdisk1 hdisk6 hdisk7
```

(물리적 공간은 동일한 볼륨 그룹 내의 한 물리적 볼륨에서 두 개의 다른 물리적 볼륨으로 이동됨)

#### ▶ 논리적 볼륨 lv02의 물리적 공간을 hdisk1에서 hdisk6으로 이동시키기

```
# migratepv -l lv02 hdisk1 hdisk6
```

## 논리적 볼륨(Logical Volume)



### Notes :

볼륨 그룹을 생성한 후에는 볼륨 그룹 내에서 논리적 볼륨을 만들 수 있다. 논리적 파티션과 논리적 볼륨은 논리적인 관점에서 만들어진다. 논리적 파티션은 물리적 파티션의 크기와 동일하다.

논리적 볼륨(LV)은 하나 혹은 여러 개의 논리적 파티션의 집합으로 이루어지며 여러 개의 PV에 걸쳐서 생성될 수 있다.

모든 PV는 같은 볼륨 그룹 내에서 확장해야 하고 특별히 크게 지정하지 않는 한, 논리적 볼륨은 최고 128개의 논리적 파티션으로 이루어진다.

하나의 볼륨 그룹 당 256개의 논리적 볼륨까지 가능하고, 대형 볼륨 그룹의 경우 512개의 논리적 볼륨까지 가능하다.



## Logical Volume의 사용

- 저널파일시스템(for example : /dev/hd4)
- 페이징 스페이스(/dev/hd6)
- 저널로그(/dev/hd8)
- 부트논리볼륨(/dev/hd5)
- raw device

### Notes :

운영체제를 설치할 때, 논리적 볼륨은 자동으로 생성된다.

논리적 볼륨을 만들 때, 운영체제에 의해서 공간은 할당되고 부트 데이터, 페이징 스페이스, 덤프 공간 같은 몇가지 대상들은 이 공간 할당 방법을 사용한다.

#### ■ JFS(Journaled File System) :

AIX의 기본 파일 시스템은 저널 파일 시스템(JFS) 이다. 이 파일 시스템은 일관성을 유지하기 위해 데이터베이스 저널링 기술을 사용한다. 사용자 파일 액세스, 명령, 어플리케이션 등은 파일 시스템 디렉토리 구조를 사용한다. 저널 파일 시스템은 /dev/hd2 논리 볼륨을 사용할 수 있다.

#### ■ Paging space :

페이징 스페이스는 실제 메모리를 초과하는 정보를 임시적으로 저장하기 위해 지정된 하드 디스크의 저장소이다. 페이징 스페이스는 /dev/hd6 논리 볼륨에 위치한다.

#### ■ Journal log :

저널 로그는 구조가 갱신될 때까지 파일 시스템의 변경된 정보가 쓰여지는 논리 볼륨이다. 저널 로그는 /dev/hd8 논리 볼륨에 있다.

#### ■ Boot Logical Volume :

부트 논리 볼륨은 부트 이미지가 들어 있는 물리적으로 연속적인 디스크 영역이다. 부트 논리 볼륨은 /dev/hd5 논리 볼륨에 있다.

#### ■ Raw Device :

raw device는 비어 있는 논리 볼륨이다. 때때로 데이터베이스와 같은 어플리케이션이 데이터를 저장하고 관리하기 위해 raw device를 필요로 한다.

## Logical Volume의 정보 리스트

```
# lsvg -l rootvg
```

LV NAME	TYPE	LPs	PPs	PVs	LV STATE	MOUNT POINT
hd5	boot	1	1	1	closed/syncd	N/A
hd6	paging	32	32	1	open/syncd	N/A
hd8	jfslog	1	1	1	open/syncd	N/A
hd4	jfs	252	252	1	open/syncd	/
hd2	jfs	210	210	1	open/syncd	/usr
hd9var	jfs	1	1	1	open/syncd	/var
hd3	jfs	2	2	1	open/syncd	/tmp
hd1	jfs	1	1	1	open/syncd	/home
hd10opt	jfs	6	6	1	open/syncd	/opt

### Notes :

#### ■ 모든 볼륨 그룹을 나열

```
# lsvg [-옵션] [볼륨그룹명]
```

- o 옵션 : 활성화 되어 있는 볼륨 그룹만을 나열
- p 옵션 : 볼륨 그룹내에 있는 물리적 볼륨(PV)의 정보를 제공
- l 옵션 : 볼륨 그룹내에 있는 로지컬 볼륨(LV)의 정보를 제공

#### ▶ 루트 볼륨그룹에 대한 정보를 제공

```
# lsvg -l rootvg
```

#### ▶ 볼륨그룹내에 있는 LV 에 관한 상태정보를 제공

```
# lslv hd1(lv이름)
```

## Mirroring

- ▣ 미러링이란 장비가 고장나는 사고가 발생하였을 때 데이터가 손실되는 것을 막기 위하여, 데이터를 하나 이상의 장치에 중복 저장하는 것
- ▣ LV를 생성할 때뿐 아니라, 이미 존재하는 LV에 대해서도 Mirroring 가능
- ▣ 사본은 서로 다른 디스크에 존재하도록 하지만, 불가피한 경우에는 같은 디스크에 존재할 수 있음

### Notes :

#### ■ 스케줄링 정책

LV의 Mirroring은 데이터의 가용성을 전제로 하므로 데이터의 사본을 만든다. 그러나, 데이터의 사본을 만들기 위해 디스크에 추가적인 쓰기 작업을 필요로 하므로 해당 LV의 성능을 저해할 수 있다.

##### ▶ 병렬 (Parallel) : 디폴트

이 정책은 논리 파티션의 모든 물리 파티션에 쓰기 작업을 동시에 시작한다. 쓰기 요청은 시간이 가장 많이 걸리는 복제가 쓰기 완료로 해야 종료된다. 읽기 작업시 응답 시간을 빠르게 하기 위해 가장 가까운 복제를 읽다.

##### ▶ 순차 (Sequential)

순차 옵션은 다음의 순서로 복제 쓰기 작업을 수행한다.

1차 → 2차 → 3차

이 방법을 선택하면 다음 복제에 쓰기 작업을 시작하기 전에 이전의 물리 파티션에 쓰기 작업이 완료되는 것을 기다린다. 읽기 작업에서는 1차 복제를 사용한다. 1차 복제를 읽을 수 없으면 사용 가능한 다음 복제를 읽다. 1차 복제는 읽기 작업을 쓰기 작업으로 변환함으로써 복구된다. 하드웨어 재배치는 물리 디바이스 드라이버를 호출할 때 지정된다.

▶ 병렬 / 순차

병렬/순차 정책은 모든 물리 파티션에 대해 쓰기 작업을 동시에 시작한다. 1차 복제를 항상 먼저 읽습니다. 1차 복제의 읽기 작업을 할 수 없는 경우에는 다음 복제를 읽다. 다음 복제에서 재시도하는 동안, LVM은 하드 디스크에 1차 복제를 재위치시켜 실패한 1차 복제를 복구한다.

▶ 병렬/라운드 로빈

병렬/라운드 로빈 옵션은 논리 파티션의 모든 물리 파티션에 쓰기 작업을 동시에 시작한다. 읽기 작업은 미리 복제들을 수시로 바꾸어 진행한다.

# Striping

- Striping은 프로세서가 하나의 디스크에서 공급하거나 또는 받아들일 수 있는 것보다 더 빠르게 데이터를 읽거나 쓸 수 있는 능력이 있음
- Striping은 하나의 LV를 여러 디스크에 분산함으로써, 그 LV의 데이터를 동시에 이용할 수 있도록 여러 디스크의 I/O가 사용되는 기법
- Striping은 성능향상을 위해 단일 파일과 같은 논리적으로 연속된 데이터 세그먼트들이 물리적으로 여러 개의 장치 즉 디스크 드라이브 등에 라운드로빈 방식으로 나뉘어 기록될 수 있는 것
- Striping은 연속적인 대용량 파일의 읽기와 쓰기 작업에 성능을 향상시킬 수 있음

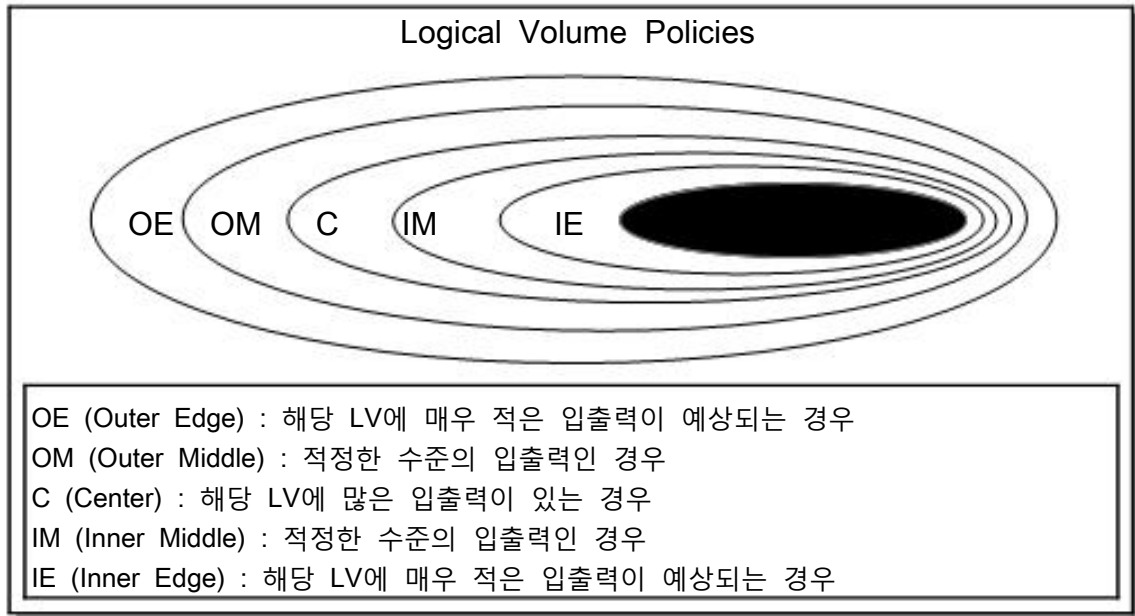
## Notes :

Striping을 사용하지 않은 LV에서, 데이터는 하나의 PP안에서의 데이터 블록 단위로 이용할 수 있다. Striping을 사용하는 LV에서는, 데이터는 stripe unit 단위로 이용할 수 있다. stripe unit의 크기는 LV가 처음 생성될 때 결정되며, 범위는 2의 제곱승으로 4KB부터 128KB까지가 가능하다.

### ■ Striping의 제한 사항

- ▶ Striping된 LV는 디스크마다 같은 수의 PP로 분산되어야 한다.
- ▶ 적어도 2개 이상의 PV가 필요하다.
- ▶ 가능한 많은 어댑터를 이용해야 한다.
- ▶ Striping을 이용하는 LV를 위한 전용 VG를 만들도록 한다.

## Logical Volume 정책



### Notes :

디스크의 헤드는 해당 디스크의 CENTER 주변에 위치하게 되는 것이 일반적이므로, 가장 비활동적인 LV들로부터 자료를 얻고자 하는 경우에는, 양쪽의 EDGE로 헤드가 움직이는데 필요한 시간은 거의 동일하게 된다. 이러한 현상은 EDGE상에 위치한 자료를 빈번히 사용하게 되거나 혹은 매우 큰 블록을 사용하는 경우의 성능에 영향을 줄 수 있다.

#### ■ interphysical과 intraphysical 정책

##### ▶ intraphysical 볼륨 할당 정책 :

PV안에서 LV의 해당 PP가 어디에 위치할 것인지를 결정한다. 선택 가능한 값은, outer edge, outer middle, center, inner middle, inner edge이다. 디스크의 center에 위치하는 PP는 가장 짧은 디스크 seek time의 특성을 가지므로 middle이나 edge에 위치한 PP의 데이터보다 빠른 액세스가 가능하다.

##### ▶ Interphysical 볼륨 할당 정책 :

하나의 LV를 구성하는 PP들을 몇 개의 PV에서 포함할 것인지를 결정한다. Interphysical 할당에서 "minimum"은 가용성을 높이기 위해 LV를 하나의 디스크의 PP만을 이용하여 만든다. maximum을 선택하면, 성능 향상을 위해 여러 개의 디스크에 PP를 분산할 수 있다.

## Logical Volume의 특성 보기

### ■ Physical Volume map ( # lslv -l lv00 )

```
lv00:/ftp
PV          COPIES      IN BAND      DISTRIBUTION
hdisk1      062:000:000  61%         000:038:024:000:000
```

### ■ Logical Partition map ( # lslv -m lv00 )

```
lv00:/ftp
LP   PP1 PV1          PP2 PV2          PP3 PV3
0001 0056 hdisk1
0002 0057 hdisk1
0003 0058 hdisk1
```

## Notes :

### ■ Physical Volume map

#### ▶ PV 열 :

논리 볼륨을 포함하는 디스크를 보여준다.  
이 경우 LV는 hdisk1에 포함되어 있다.

#### ▶ COPIES 열 :

각각의 미러에 대한 해당 디스크에 있는 물리 파티션(PPs)의 개수를 보여준다.  
001는 첫번째 복제에 있는 1 PP이며 000:000는 디스크의 두 번째와 세 번째 복제에는 PPs가 없다는 것을 보여준다.

#### ▶ IN BAND 열 :

100%는 LV 생성시 정의한 정책에 맞게 모든 PP들의 배치가 이뤄졌다는 것을 의미한다.

#### ▶ DISTRIBUTION 열 :

PP(Physical Partition) 분포이다.

이것은 바깥 끝(왼쪽)에 안쪽 끝(오른쪽)까지의 PP의 분포를 보여준다.

■ Logical Partition map :

LP 0001은 hdisk1의 PP 0219에 대응함을 알 수 있다.  
만약 mirroring된 LV일 경우에는, PP2, PV2, PP3, PV3의 정보도 보여준다.  
화면으로 보여지는 LV의 경우는 mirroring된 LV가 아니므로, 이들 항목에는 아무 내용도 없다.

■ 논리적 볼륨 정보를 표시하는 방법

# `lslv [ -L ] [ -l|-m ] [ -nPhysicalVolume ] LogicalVolume`

-L 옵션 : 볼륨 그룹에서의 잠금을 확보하기 위한 대기를 지정

-l 옵션 : 논리적 볼륨(LV)에서 각 물리적 볼륨(PV)에 대한 다음 필드를 나열

-m 옵션 : 각 논리적 공간(LP)에 대한 다음 필드를 나열

▶ 물리적 볼륨에 의해 논리적 볼륨 lv03에 대한 정보를 표시하기

# `lslv -l lv03`

■ 논리적 볼륨 할당 맵을 표시하는 방법

# `lslv [ -L ] [ -nPhysicalVolume ] -pPhysicalVolume [ LogicalVolume ]`

-p 옵션 : 변수에 대한 논리적 볼륨(LV) 할당 맵을 표시

-n 옵션 : 변수의 특정 서술어 영역으로부터 정보를 액세스

▶ hdisk0에 대한 논리적 볼륨(LV) 할당 맵을 표시하기

# `lslv -p hdisk0`



## Logical Volume의 추가

# smit mklv

논리적 볼륨 추가			
입력 필드에 값을 입력하거나 선택하십시오. 원하는 것을 모두 변경한 후, Enter 키를 누르십시오.			
[맨위]	[입력 필드]		
논리적 볼륨 이름	<input type="text"/>		
* 볼륨 그룹 이름	rootvg		
* 논리적 공간 수	<input type="text"/>		#
물리적 볼륨 이름	<input type="text"/>		+
논리적 볼륨 유형	<input type="text"/>		
물리적 볼륨 상의 위치	middle		+
물리적 볼륨의 범위	minimum		+
할당에 사용할 물리적 볼륨의 최대 갯수	<input type="text"/>		#
각 논리적 공간의 사본 수	1		+
미러 쓰기 일치성?	예		+
각각의 논리적 공간 사본을 별도의 물리적 볼륨 상에 할당하시겠습니까?	yes		+
[추가...10]			
F1=도움말	F2=화면갱신	F3=취소	F4=리스트
F5=재설정	F6=명령	F7=편집	F8=이미지
F9=헬프	F10=종료	Enter=실행	

### Notes :

#### ■ 논리 볼륨의 추가 메뉴

이 메뉴에서 LV의 이름을 지정할 수 있으며, 지정하지 않을 경우에는 시스템에서 자동으로 이름이 생성된다. LV의 크기(파티션의 개수로)를 반드시 지정해야 하므로, LV를 추가할 VG의 파티션에 대한 정보를 알고 있어야 한다.

#### ■ 디폴트 4MB의 파티션을 사용하는 경우

##### ▶ 논리적 공간 수 항목 :

1 을 설정하면 논리 볼륨에 4MB 공간을 할당한다.

물리적 볼륨 상의 위치 항목 : 특정 디스크를 지정하고 해당 디스크에서의 위치 (inner middle, center 등등)를 지정하도록 한다.

- ▶ 물리적 볼륨의 범위 항목 : LV가 어떻게 할당될 것인지를 정의한다.  
Minimum 값은 LV를 가능한 한 하나의 PV에서 생성되도록 한다. Maximum 값은 해당 VG에 포함된 가능한 많은 PV에 분산되어 생성되도록 한다.
- ▶ 할당에 사용할 물리적 볼륨의 최대 갯수 항목 :  
물리 볼륨에 대한 내부 정책(intrapolicy) 및 상호 정책(interpolicy)을 결정한다.  
이는 LV를 최대 몇 개의 디스크에 걸쳐 생성할 것인지를 설정한다.

#### ■ 논리적 볼륨 작성 명령어

# mklv [-옵션]

- a 옵션(Position) : 물리적 볼륨 내 할당 방침을 설정
- c 옵션(copies) : 각각의 논리적 공간에 할당된 물리적 공간의 수를 설정
- e 옵션(Range) : 물리적 볼륨간 할당 방침을 설정
- S 옵션(StripeSize) : 스트립 당 바이트 수를 지정
- y 옵션(logicalvolume\_name) : 시스템 생성 이름 대신 사용하기 위한 논리적 볼륨 이름을 지정

- ▶ vg02 볼륨 그룹에 하나의 논리적 공간과 총 두 개의 데이터 사본이 있는 논리적볼륨을 만들기

```
# mklv -c 2 vg02 1
```

- ▶ vg03 볼륨 그룹에 아홉 개의 논리적 공간과 최대 두 개의 물리적 볼륨 전체에 분포하는 총 세 개의 사본을 가지며, 할당 방침이 엄격하지 않은 논리적 볼륨 만들기

```
# mklv -c 3 -u 2 -s n vg03 9
```

## Logical Volume의 삭제

# smit rmlv

논리적 볼륨 제거

입력 필드에 값을 입력하거나 선택하십시오. 원하는 것을 모두 변경한 후, Enter 키를 누르십시오.

논리적 볼륨 이름

[입력 필드]

+

F1=도움말	F2=화면갱신	F3=취소	F4=리스트
F5=재설정	F6=명령	F7=편집	F8=이미지
F9=셸	F10=종료	Enter=실행	

### Notes :

■ 논리 볼륨의 삭제 메뉴

rmlv 명령어를 이용하여 LV를 삭제 할 수 있다.

■ 논리 볼륨 삭제 명령어

# rmlv [ -B ] [ -f ] [ -p Physical Volume ] LogicalVolume

-B 옵션 : 논리적 볼륨이 -l 플래그를 사용하여 작성된 경우, 상위 논리적 볼륨에 대해 chlvcopy -B -s를 실행함

-f 옵션 : 확인을 요청하지 않고 논리적 볼륨을 제거

-p 옵션 : PhysicalVolume의 논리적 공간(LP)만을 제거

▶ 사용자 확인 없이 lv05 논리적 볼륨 제거하기

# rmlv -f lv05

※ 참고

Journaled File System이나 페이징 공간의 용도로 쓰이고 있는 LV들은 rmlv명령어를 사용하지 마시오.

## Unit 6. Working with File Systems

---

File Systems 란?  
File Systems의 필요성  
AIX의 File Systems  
유닉스 File System  
Inode의 구조  
File Systems의 단편화  
JFS와 JFS2비교  
파일 및 File Systems 사이즈  
File Systems 압축  
JFS 로그  
현 File Systems 상태  
/etc/filesystems  
마운트된 File Systems 보기  
이전 논리볼륨에 JFS 추가  
JFS 추가하기  
File Systems 마운트  
빈 디렉토리 마운트 하기  
파일이 있는 디렉토리 마운트  
JFS 속성 보기 및 변경  
JFS 삭제 하기

## File Systems 란?

■ File System : 데이터를 저장하며, 계층형 디렉토리 구조

■ AIX에서 지원하는 File Systems : JFS, Cdrfs, Nfs

- ▷ JFS : File System의 기본 유형으로 데이터 베이스 저널링 기술을 사용한다. 이 저널링 기술 사용으로 File System이 비정상적으로 정지될 때 File System이 손상되는 것을 막아준다.
- ▷ Cdrfs : 정상 File System 인터페이스(열기, 읽기 및 닫기)를 통한 CD-ROM 내용 액세스를 허용하는 File System 유형이다.
- ▷ Nfs : 원격 시스템에 있으면서 국지 시스템에 파일이 상주하는 것처럼 액세스하는 File System 유형이다.

### Notes :

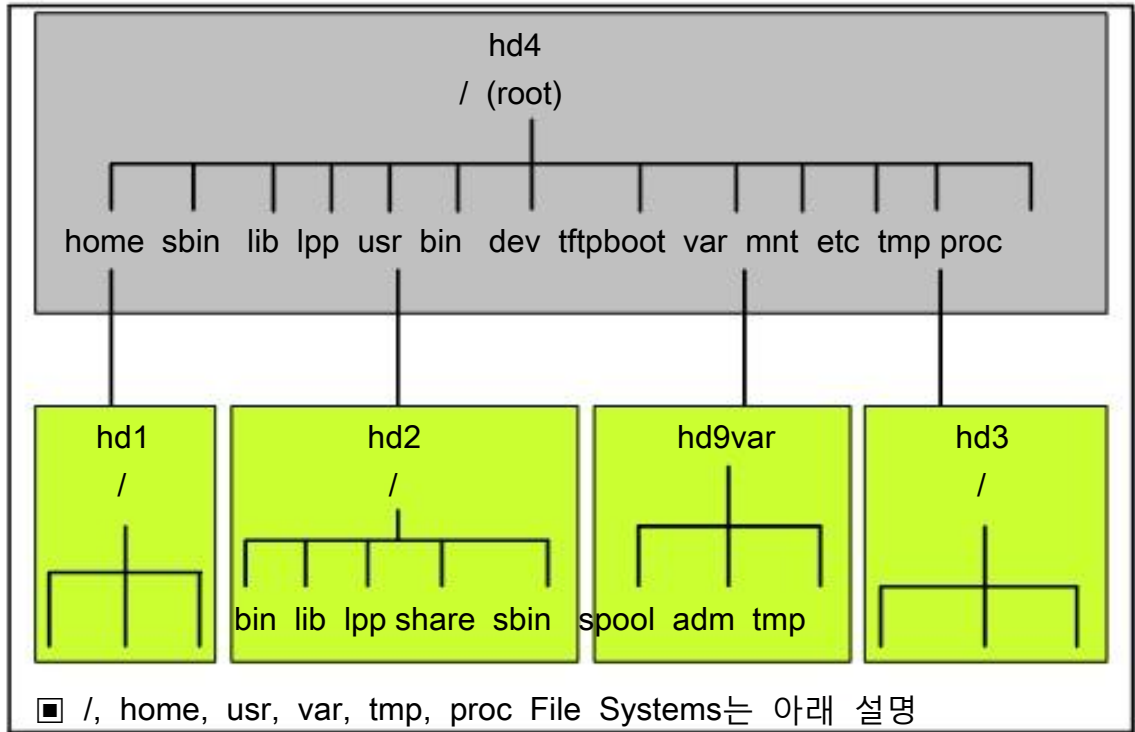
파일 시스템은 데이터를 저장하는 계층형 디렉토리 구조이다. 일반적으로, 각 파일 시스템은 관련된 파일과 디렉토리를 모아놓고 있다. 파일 시스템이 하나의 루트와 함께 마치 하나의 파일 트리처럼 모여 있다.

파일 구조는 사용자가 데이터를 찾고 탐색할 수 있도록 지도를 제공하고 있다. 시스템 관리자는 각 파일 시스템에 적당한 디스크 공간을 할당한다. 만약 파일 시스템 공간이 부족하면, file full 오류가 발생하면서, 더 이상의 데이터가 파일 시스템에 추가되지 않는다.

만약 파일 시스템이 공간을 더 필요로 하면, 시스템 관리자는 파일 시스템에 저장장치를 할당할 수 있다. 관리자가 디스크 자원을 제어하도록 허락하는 것이 바로 파일 시스템(File System)이다.

만약 AIX를 처음으로 설치했다면, 기본적으로 시스템은 /, /var, /tmp, /proc, /usr, /home 를 설치한다. 이 파일 시스템들은 AIX에서 각각의 목적을 갖고 있다.

## AIX의 File Systems



### Notes :

■ 루트(/) 파일 시스템 :

루트(/) 파일 시스템은 논리 볼륨 /dev/hd4에 마운트 된다. 이는 계층 파일 트리의 맨 위에 위치한다. 루트 파일 시스템은 부팅을 위한 디바이스 정보와 프로그램을 포함하며, 시스템 운영을 위해 중요한 파일과 디렉토리를 포함한다. 다른 디폴트 파일 시스템의 마운트 포인트는 이 파일 시스템 하부에 위치한다.

■ /home 파일 시스템 :

/home 파일 시스템은 논리 볼륨 /dev/hd1에 마운트 된다. 여기에는 홈 디렉토리 및 사용자 파일 및 디렉토리가 저장된다..

■ /usr 파일 시스템 :

/usr 파일 시스템은 논리 볼륨 /dev/hd2에 마운트 된다. 여기에는 운영 체제 명령, 라이브러리, 그리고 응용 프로그램이 들어 있다. 동일한 하드웨어 구조를 가진 시스템과 공유할 수 있는 파일도 이 파일 시스템에 위치한다.

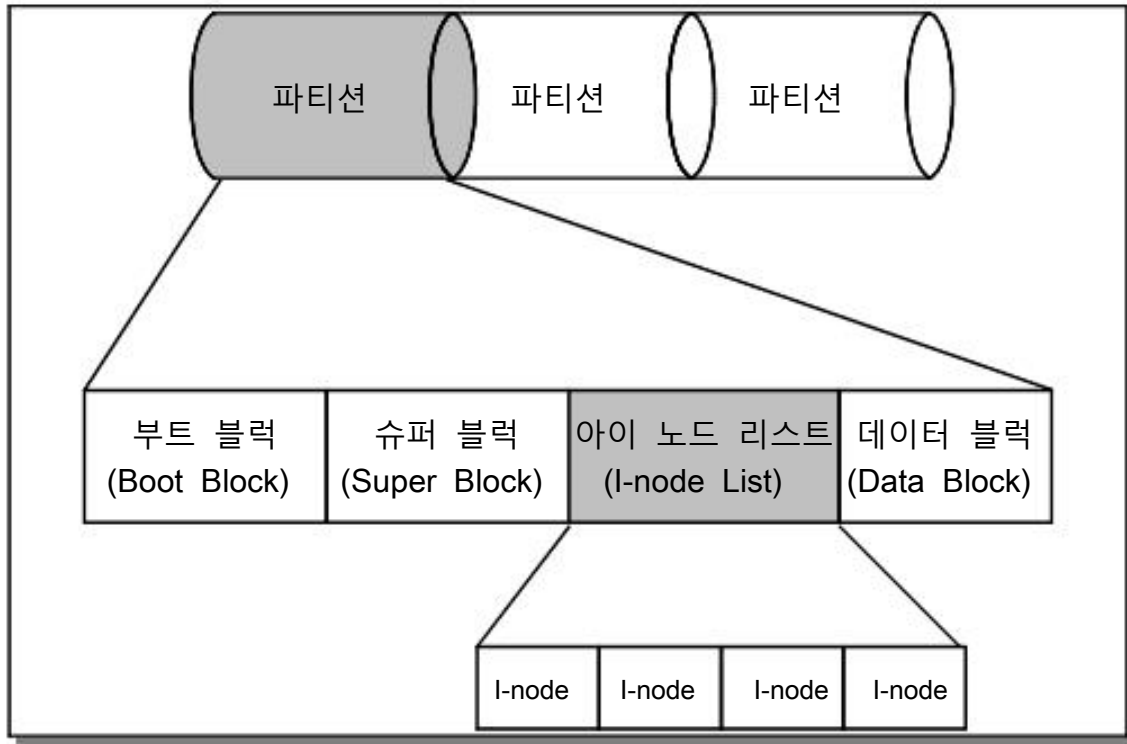
■ /tmp 파일 시스템 :

/tmp 파일 시스템은 논리 볼륨 /dev/hd3에 마운트 된다. 이 파일 시스템은 시스템이 생성한 임시 파일을 포함한다. 디폴트로, /tmp 디렉토리는 비어 있는 디렉토리지만, 사용자는 임시로 파일을 저장할 수 있다. 시스템은 이 공간을 시스템 작업을 위한 버퍼로 사용하기 때문에 주기적으로 정리해 주어야 한다.

■ /proc 파일 시스템 :

/proc 파일 시스템은 AIX 버전 5 이상에서 지원하는 의사 파일 시스템이다. 이 파일 시스템은 프로세스와 커널 데이터 구조를 대응하는 파일에 대응시킨다. 이 파일 시스템에는 각 프로세스마다 프로세스 ID와 똑같은 이름을 가진 디렉토리가 지정되어 있다. 이 디렉토리에는 내부 프로세스 제어 데이터 구조에 대응하는 여러 개의 파일과 서브 디렉토리가 생성된다. 이 파일 가운데 일부는 읽기 전용이며, 나머지는 프로세스 제어용으로 사용될 수 있다.

## 유닉스 File System



### Notes :

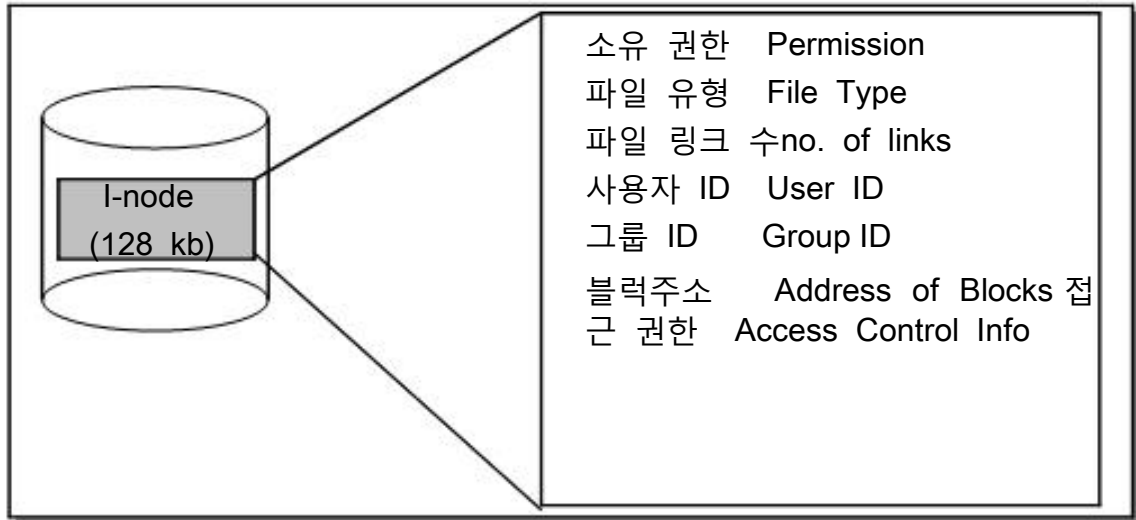
File System은 파일과 디렉토리의 계층적 구조 (파일트리)이다.

또한, 파일 시스템은 정보를 유지하고, 파일 또는 디렉토리 데이터 위치를 알려준다. File Systems는 Boot Block, Super Block, Inode 그리고 Data Block 로 구성되어 있다.

- Boot Block : File Systems의 시작부에 위치하며, 첫번째 sector를 차지한다.
- Super Block : File Systems에 관한 전체적인 정보들을 저장하는 블록이다.
- Inode : 64byte의 크기를 가지며, 하나의 어떤 파일이나 디렉토리를 생성시 그에 대한 정보가 저장되어 있다.
- Data Block : 실제로 데이터가 저장되는 블록 이다.



## Inode의 구조



### Notes :

i-노드는 디스크의 그 데이터 위치 등을 포함하여 파일에 대한 정보를 담고 있다. 디렉토리는 파일이름과 i-노드를 가지고 있다. 특정 파일을 찾기 위해 시스템은 디렉토리를 참조하여 i-노드 번호를 확보한다. i-노드 번호를 찾으면 i-노드가 디스크 위치 정보를 제공해준다.

i-노드 크기는 128 bytes로 되어 있다. 그리고 그 안에는 디렉토리에 저장된 파일이름과 데이터 블록에 저장되어 있는 실제 파일의 내용을 제외한 다른 특성 정보들이 저장된다.

i-노드는 NBPI(Number of bytes per i-node)에 의해 그 숫자가 정해진다.

NBPI(Number of bytes per i-node) : inode 당 바이트 수

#### ■ i-노드 내용 보기

ls-li 명령어를 이용하여 i-노드 내용을 볼 수 있다.

```
# opmon[/home/opmon/omg]ls -li
```

```
total 59
```

```
6200 -rw-r--r--      1 opmon  staff      86563 Aug 07 01:41 smit.log
```

```
6201 -rw-r--r--      1 opmon  staff      7859 Aug 06 17:31 smit.script
```

## JFS와 JFS2비교

	JFS	JFS2
최대 파일사이즈	64 Gigabytes	4 Petabytes
최대 File Systems 사이즈	1 Terabyte	4 Petabytes
i-node 사이즈	128 bytes	512 bytes
i-node 수	고정	유동
프래그먼트 / 블록사이즈	512 / 4096 byte	512 / 4096 byte

### Notes :

JFS2 의 새로운 기능

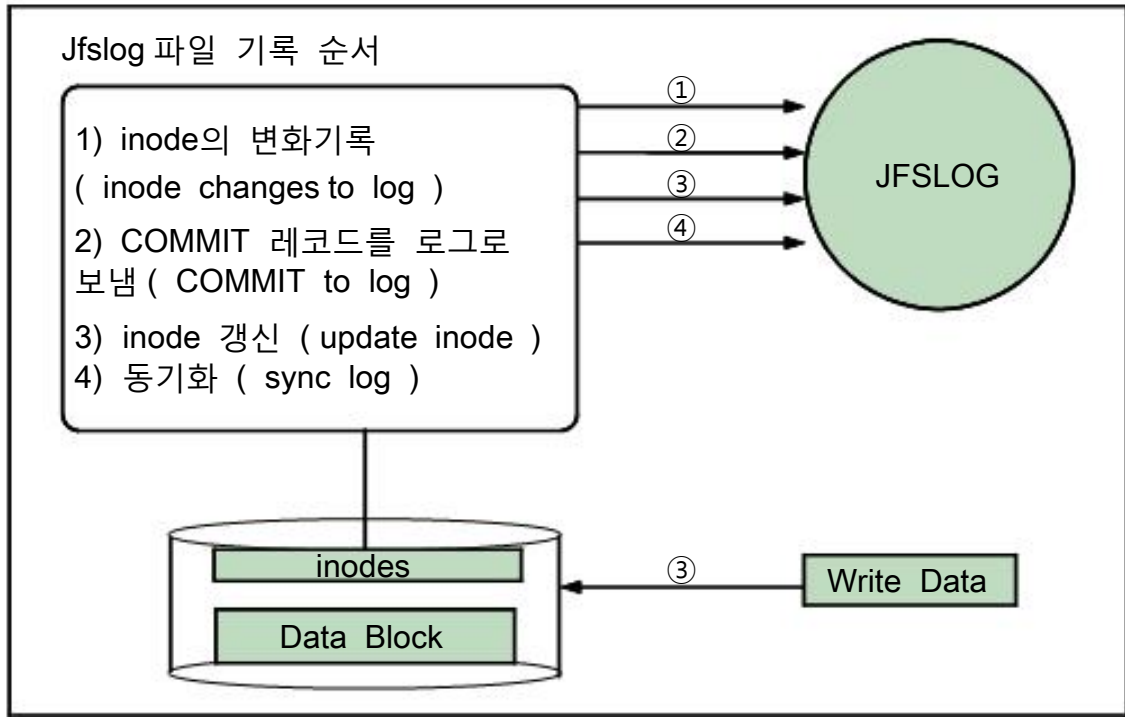
1. Journaling
2. 다양한 블록 사이즈
3. 동적인 디스크 inode 할당
4. 디렉터리 구조
5. 명령어와 유틸리티 및 호환성
6. 시스템 웹 관리도구(Web-based System Manager)
7. file system 점검과 복구

### ■ 참고

용량 단위

1 byte	8 bit
1 KB	1024 byte
1 MB	1,048,576 byte / 1024 KB
1 GB (Gigabytes)	1,073,741,824 byte / 1024 MB
1 TB (Terabyte)	1,099,511,627,776 byte / 1024 GB
1 PB (Petabytes)	1,125,899,906,842,624 byte / 1024 TB

## JFS로그



### Notes :

#### ■ Jfslog 파일 기록 순서

1. jfslog는 i- 노드의 변화를 로그에 기록한다.

Jfslog는 요청된 활동과 수행된 명령어에 대한 정보를 기록한다.

2. jfslog는 COMMIT 레코드를 로그로 보낸다.

이는 jfslog가 첫번째 단계에서 정보의 로깅을 완료했음을 나타낸다.

3. 시스템은 데이터를 쓰고 디스크의 i-node를 갱신한다.

4. Sync 시점을 쓴다.

가장 최근의 sync 레코드는 시스템에 명령에 대한 작업이 완료되었음을 알린다.  
jfslog도 시스템이 디스크 쓰기를 완료하고 i-node를 썼음을 기록한다.

AIX는 journaled file system을 사용하는데, 이것은 데이터를 쓸 때 transaction 로그에 i-노드를 저장하는 것을 의미한다. 이를 통해 File Systems의 무결성을 유지한다.  
journaled file system은 시스템 crash가 발생했을 때 빨리 재시동될 수 있도록 해 준다.  
과거에 journaled file system 을 사용하지 않았을 때는 재시동시 시스템 검사로 인해 몇 시간이 소요되기도 했지만 현재 JFS와 JFS2를 이용하면 빠른 재시동이 가능하다.

## 현 File Systems 상태

# lsfs

이름	노드명	마운트 지점	VFS	크기	옵션	자동	사용통계
/dev/hd4	--	/	jfs	196608	--	예	아니오
/dev/hd1	--	/home	jfs	65536	--	예	아니오
/dev/hd2	--	/usr	jfs	6094848	--	예	아니오
/dev/hd9var	--	/var	jfs	524288	--	예	아니오
/dev/hd3	--	/tmp	jfs	655360	--	아니오	아니오
/dev/cd0	--	/cdrom	cdrfs	--	ro	아니오	아니오
/dev/lv00	--	/oracle	jfs	10616832	rw	예	아니오
/dev/lv01	--	/weblogic	jfs	6946816	rw	예	아니오
/dev/lv03	--	/summa	jfs	4128768	rw	예	아니오
/dev/lvnetbk01	--	/netbackup	jfs	655360	rw	예	아니오
/dev/lv02	--	/usr1	jfs	5242880	rw	예	아니오

### Notes :

lsfs 명령어 : 현재 File Systems를 보는 명령어

#### ■ lsfs 명령어 수행결과

- ▶ 이름 (Name) : LV, CD-RO장치, Remote 디렉토리 이름
- ▶ 노드명 (Nodename) : NFS일 경우만 사용되는 원격 시스템 이름
- ▶ 마운트 지점 (Mount Pt) : 마운트 포인트 디렉토리
- ▶ 가상 File Systems (VFS) : 가상 파일 시스템 타입 512-byte 블록
- ▶ 크기 (Size) : 512-byte블록
- ▶ 옵션 (Options) : /etc/filesystem 에 있는 마운트 옵션
- ▶ 자동 (Auto) : /etc/filesystems 에 있는 시스템 시작시 마운트 하라는 뜻

#### ■ 참고

smit(system management interface tool) 으로는

smit fs → File Systems → List All File Systems 도 동일한 결과이다.

## /etc/filesystems

# vi /etc/filesystems

```
/:
    dev          = /dev/hd4
    vfs           = jfs
    log           = /dev/hd8
    mount         = automatic
    check         = false
    type          = bootfs
    vol           = root
    free          = true

/home:
    dev          = /dev/hd1
    vfs           = jfs
    log           = /dev/hd8
    mount         = true
    check         = true
    vol           = /home
    free          = false
```

### Notes :

■ /etc/filesystems 파일 : 파일 시스템의 레이아웃 특징, 속성등을 담고 있다.

/ 자원이름 :            속성 = 값

- ▶ check : fsck 명령어로 검사될 파일 시스템인지 아닌지를 설명해 준다.
- ▶ dev : 파일 시스템이 있는 논리 볼륨에 대한 정보를 포함한다.
- ▶ mount : 디폴트로 파일 시스템을 마운트 해야 할 지를 결정하기 위해 mount 명령을 사용한다. 사용 가능한 값은 다음과 같다.
  - Automatic : 시스템 시작 시에 파일 시스템이 자동으로 마운트 된다.
  - True : 파일 시스템이 mount all 명령으로 마운트 된다. 이 명령은 시스템 초기화 하는 동안 파일 시스템을 자동으로 마운트 할 때 수행된다.
  - False : 파일 시스템이 자동으로 마운트 되지 않는다.
- ▶ type : 파일 시스템을 어떤 그룹으로 처리할지 설명해준다.
- ▶ vfs : 파일 시스템의 유형을 명시한다.
- ▶ log : 볼륨 그룹내의 파일 시스템 구조 정보를 저장하기 위한 로그 장치 이름

## 모든 마운트된 File Systems 보기

# mount

노드	마운트됨	마운트됨	vfs	날짜	옵션
/dev/hd4	/		jfs	6월 15일 19:17	rw,log=/dev/hd8
/dev/hd2	/usr		jfs	6월 15일 19:17	rw,log=/dev/hd8
/dev/hd9var	/var		jfs	6월 15일 19:17	rw,log=/dev/hd8
/dev/hd3	/tmp		jfs	6월 15일 19:17	rw,log=/dev/hd8
/dev/hd1	/home		jfs	6월 15일 19:18	rw,log=/dev/hd8
/dev/lv00	/oracle		jfs	6월 15일 19:18	rw,log=/dev/hd8
/dev/lv01	/weblogic		jfs	6월 15일 19:18	rw,log=/dev/hd8

### Notes :

mount 명령어 : 마운트 되어있는 File Systems를 보여줌

#### ■ lsfs 명령어 수행결과

- ▶ 노드 (node) : 원격에서 마운트가 되었다면 그 노드 이름
- ▶ 마운트장치 (mounted) : 마운트된 장치 이름
- ▶ 마운트 포인트 (mounted over) : 마운트 포인트
- ▶ File Systems 유형 (vfs) : 가상 파일 시스템 유형
- ▶ 날짜 (date) : File Systems가 마운트된 시간
- ▶ 옵션 (options) : rw는 read/write를 의미하고 로그는 해당 파일 시스템

#### ■ 참고

smit(system managerment interface tool) 으로는

smit fs → File system → List All Mounted File Systems 도 동일한 결과이다.

## JFS 추가하기

# smit crjfsstd

표준 저널 File Systems 추가

입력 필드에 값을 입력하거나 선택하십시오. 원하는 것을 모두 변경한 후, Enter 키를 누르십시오.

	[입력 필드]	
볼륨 그룹 이름	testvg	
* File Systems 크기(512바이트 블록)	1	#
* 마운트 위치	/	
시스템 재시작시 자동으로 마운트하시겠습니까?	아니오	+
사용권한	읽기/쓰기	+
마운트 옵션		+
디스크 사용통계를 시작하시겠습니까?	아니오	+
프래그먼트 크기(바이트)	4096	+
inode 당 바이트 수	4096	+
그룹 크기(메가바이트) 할당	8	+

F1=도움말

F5=재설정

F9=헬프

F2=화면갱신

F6=명령

F10=종료

F3=취소

F7=편집

Enter=실행

F4=리스트

F8=이미지

### Notes :

smit crjfsstd : jfs 추가 하기 위한 smit fast path

- 볼륨 그룹이름 (Volume Group Name) :  
추가 하기 원하는 볼륨 그룹을 선택 또는 직접 넣을 수 있다.
- File Systems 크기 (Size of file system) :  
원하는 크기를 지정

## File Systems 마운트

# smit mount

파일 시스템 마운트

입력 필드에 값을 입력하거나 선택하십시오. 원하는 것을 모두 변경한 후, Enter 키를 누르십시오.

	[입력 필드]	
파일 시스템 이름	<input type="text"/>	+
마운트될 등록부	<input type="text"/>	+
파일 시스템 유형		+
강제로 마운트합니까?	아니오	+
마운트할 파일 시스템이 들어 있는 원격 노드	<input type="text"/>	
제거가능한 파일 시스템으로 마운트합니까?	아니오	+
읽기 전용 시스템으로 마운트합니까?	아니오	+
이 마운트를 통해 장치 접근을 허가하지 않습니까?	아니오	+
이 파일 시스템에서 sgid 프로그램과 SUID 실행을 허가하지 않습니까?	아니오	+

### Notes :

smit mount : File Systems를 마운트 하는 smit fast path

- 파일 시스템이름 (FILE SYSTEM name) : 마운트 할 File Systems 이름을 넣는다.
- 마운트 될 등록부 ( DIRECTORY over which to mount) : 마운트 할 곳을 지정
- File Systems 유형 (TYPE of file system) : File Systems 유형 선택
- 강제로 마운트(FORCE the mount?) : yes 또는 no를 지정할 수 있다.

- 파일 시스템을 사용가능하게 만들기

# mount[ -t Type ]

-t 옵션 : Type type=Type 속성을 포함하며 마운트되지 않은 /etc/filesystems 파일의 모든 스탠자를 마운트

- ▶ /etc/filesystems 파일로부터 특정 유형의 파일이나 디렉토리를 마운트하기
- # mount -t remote



## JFS File Systems 삭제 하기

# smit rmfs

저널 파일 시스템 제거

입력 필드에 값을 입력하거나 선택하십시오. 원하는 것을 모두 변경한 후, Enter 키를 누르십시오.

[입력 필드]

* 파일 시스템 이름		+
마운트 위치 제거	아니오	+

### Notes :

smit rmfs 명령어 : jfsFile Systems를 삭제 하기 위한 smit fast path

파일 시스템이 JFS(저널 파일 시스템)이면, rmfs 명령은 파일 시스템이 상주하는 논리적 볼륨과 /etc/filesystems 파일에 있는 관련 스탠자를 둘 모두 제거한다. 파일 시스템이 JFS 파일 시스템이 아니면, 명령은 /etc/filesystems 파일의 관련 스탠자 만 제거한다.

FileSystem 매개변수는 제거될 파일 시스템을 지정한다.

#### ■ 파일 시스템을 제거한다.

# rmfs [ -r ] FileSystem

-r 옵션 : File Systems 마운트 지점을 제거

- ▶ /test 파일 시스템, /etc/filesystems 파일의 항목과 그것이 존재하는 논리적 볼륨을 제거하기

# rmfs /test

## Unit 7. Managing File Systems

---

File Systems 공간 관리  
디스크 공간 확인  
증가하는 파일관리  
디스크 사용량 확인  
파일제거를 위한 명령어  
File Systems의 단편화 제거  
File Systems 검증

## FileSystems 공간 관리

- ▣ 파일 시스템 사용량 증가를 모니터링
- ▣ 사용량 증가의 원인을 판단
- ▣ 증가하는 파일을 관리
- ▣ 파일 시스템 공간 사용을 관리
- ▣ 디스크 쿼터(Disk Quota)를 이용하여 사용자 디스크 사용을 제어
- ▣ 파일 시스템의 단편화 제거
- ▣ 파일 시스템의 무결성을 검증
- ▣ 파일 시스템의 설정을 문서화

### Notes :

파일 시스템이 꽉 차게 되었을 때 자동으로 파일 시스템의 크기가 증가하지는 않는다. 그러나, 시스템 관리자는 동적으로 파일 시스템의 크기를 증가시킬 수 있다. 파일 시스템을 지속적으로 모니터링하여, 꽉 차기 전에 SMIT(system management tool)을 이용하여 파일 시스템을 확장하는 것이 시스템 관리자의 역할이다.

#### ■ 시스템 경고

일반적으로, 파일 시스템의 용량이 모두 찼음을 알려 주는 경고는 없다. 단지 사용자가 파일 시스템에 더 많은 데이터 쓰기를 시도했고, 그 때 파일 시스템이 완전히 꽉 차있는 경우, 파일 시스템이 모두 찼음을 말해주는 메시지를 받게 된다.

이는 사용자에게 예상치 못한 결과를 초래할 수 있다. 데이터 블록을 다 사용했다 라 도, 여전히 디렉토리의 공간은 있을 수 있다. 이러한 경우에 파일은 생성할 수 있지만, 데이터 블록이 가용하지 않기 때문에 그 파일은 0 바이트의 크기를 갖게 된다. 이러한 경우, 용량 산정은 시스템 관리자가 문제를 미리 피하도록 도움을 줄 수 있다.

※ 디스크쿼터 : 사용자가 이용할 수 있는 하드디스크의 양을 제한하는 것

## 디스크 공간 확인

# df

File Systems	512 블록수	Free	%Used	lused	%lused	마운트 위치
/dev/hd4	196608	50560	75%	3562	8%	/
/dev/hd2	6094848	1133672	82%	88229	12%	/usr
/dev/hd9var	524288	117880	78%	1454	3%	/var
/dev/hd3	655360	353024	47%	656	1%	/tmp
/dev/hd1	65536	32880	50%	1079	14%	/home
/dev/lv00	10616832	381296	97%	11089	1%	/oracle
/dev/lv01	6946816	1205952	83%	109150	13%	/weblogic
/dev/lv03	4128768	3699152	11%	3096	1%	/summa
/dev/lvnetbk01	655360	619008	6%	108	1%	/netbackup

### Notes :

df 명령어 : 파일 시스템의 공간에 대한 정보

#### ■ 사용되지 않은 디스크 공간의 목록 보기

파일 시스템 증가를 모니터링 하기 위해 df 명령어를 사용한다. 이 명령어는 마운트 되어 있는 모든 파일 시스템의 전체 공간과 사용되지 않은 공간에 관한 정보를 보여 준다.

#### ■ 파일 시스템의 공간에 대한 정보

# df [options]

-l 옵션 : 사용된 공간의 양을 보여준다.

-k 옵션 : 사용되지 않은 공간을 1 KB 블록 단위로 보고한다

▶ 현재 디렉토리가 위치해 있는 파일 시스템상의 사용 가능한 공간을 표시하기  
# df .

Device	512-blocks	free	%used	lused	%lused	Mounted on
/dev/hd4	19368	9976	48%	4714	5%	/

## 증가하는 파일관리

### ■ /var/adm/wtmp :

로그인 또는 시스템 관련 정보를 포함

### ■ /var/spool/\*/\* :

사용자별 메일 관련 파일 및 crontab 설정, 프린트 등의 파일을 포함

### ■ smit.log :

방문했던 smit 화면, aix 명령 그리고 결과에 대한 기록을 포함

### ■ smit.script :

smit에 의해서 실행된 aix 명령을 포함

### ■ /etc/security/failedlogin :

성공하지 못한 로그인 시도의 목록

### ■ /var/adm/sulog :

superuser로 변경하기 위한 시도 목록 및 su명령어의 기록

## Notes :

용량 산정의 또 다른 부분은 크기가 증가하는 파일의 관리다. 증가하는 파일은 모니터링 해야 하고 주기적으로 정리해 주어야 한다. 관리할 수 없을 정도로 증가하는 일반적인 유형의 파일들이 있는데, 시스템 관리자는 이러한 파일들을 지속적으로 관리하여 파일 시스템이 꽉 차지 않도록 하고 싶을 것이다.

위에 나열한 파일들은 아주 크게 증가하는 경향이 있어, 주기적으로 정리를 해줄 필요가 있다. 만약 이러한 파일이 스스로 재생성 된다는 것을 확신할 수 없다면, 해당 파일들을 제거하지 말고, touch 명령어로 파일들을 재생성 하더라도, 그 파일들은 정확한 소유권과 사용 권한을 갖도록 재생성 되지 않을 수도 있다. 널 파일의 방향을 재지정을 하여 내용을 제거하는 것이 보다 안전하다.

### ■ /var/adm/wtmp

이 파일은 시스템에 대한 모든 로그인 목록을 포함한다. 이 파일을 읽기 위해서는 다음의 명령어를 사용한다.

```
# who /var/adm/wtmp
```

이 파일을 정리하기 전에 이 파일의 내용을 저장하고 싶으면, 아래 명령어를 사용하여 결과를 다른 파일에 방향 재지정 하면 된다.

```
# who /var/adm/wtmp > newfile
```

`/var/adm/wtmp` 파일은 로그인 활동에 관한 히스토리 데이터를 포함하고 있어 필요한 파일이다. 이 파일은 항상 몇 일간의 로그인 활동 정보를 포함하고 있어야 한다.

#### ■ `/var/spool/*/*`

`/var/spool/*/*` 파일은 메일과 사용 중에 크기가 증가하는 파일 뿐만 아니라 프린터 파일을 포함한다. 증가를 모니터링 하기 위해서는, 지속적으로 오래된 파일에 대한 디렉토리를 주시해야 한다.

#### ■ `smit.log`

`smit.log` 파일은 SMIT(system management tool)이 사용될 때 크기가 빠르게 증가한다. SMIT(system management tool)을 사용해서 `mksysb` 백업을 할 때, `smit.log` 파일은 아주 커져서 작업 후에는 크기를 줄여주거나 지워야 한다.

`smit.log` 파일은 자동으로 재생성 되기 때문에 삭제할 수 있다. `smit.log`가 루트 파일 시스템에 있으면, 그 파일이 루트 파일 시스템을 빨리 채워서 시스템 손상을 초래할 수 있기 때문에 주기적으로 지워주어야 한다.

/ (루트) 파일 시스템을 살펴보려면, 다음의 명령어가 유용한 방법이다.  
# `du -x` : 디스크에 사용된 블록 수 표시

#### ■ `smit.script`

`smit.script` 파일도 자동으로 재생성 되기 때문에 삭제가 가능하다. `smit.script`가 루트 파일 시스템에 있으면, 그 파일이 루트 파일 시스템을 빨리 채울 수도 있어 시스템 손상을 초래할 수 있기 때문에 정기적으로 삭제해야 한다

#### ■ `/etc/security/failedlogin`

이 파일은 성공하지 못한 로그인 시도의 목록을 포함한다. `who` 명령어를 사용해서 이 파일을 읽을 수 있다. 해당 내용을 삭제하려면 다음의 명령어를 사용하면 된다.  
# `cat /dev/null > /etc/security/failedlogin`

#### ■ `/var/adm/sulog`

`sulog` 파일은 시스템에서 수행된 모든 `su` 명령어의 목록을 포함한다. `su` 활동에 대한 일종의 감시 트레일이다.

## 디스크 사용량 확인

```
# du -a /home/opmon/tak
```

디스크 블록수	파일명
14	/home/opmon/tak/lsvg.txt
23	/home/opmon/tak/mklv.txt
6	/home/opmon/tak/rmlv.txt
7	/home/opmon/tak/syncvg.txt
11	/home/opmon/tak/mklvcopy.txt
5	/home/opmon/tak/rmlvcopy.txt

### Notes :

파일 시스템 사용이 증가하는 경우, 어느 파일이 공간을 많이 차지하는지 찾는 데에 du(disk usage) 명령어가 유용하다.

du 명령어를 옵션 없이 사용하면, 512 바이트 블록의 단위로 디렉토리별로 디스크 사용의 합계를 재귀적으로 보여준다. 때로 이 결과를 sort 명령어로 파이프하면 더 큰 숫자를 빨리 보는 데 도움이 된다.

```
# du | sort -rn
```

#### ■ 디스크 사용법 요약

```
# du [ -x ]
```

-x 옵션 : 특정 파일 시스템을 주목할 할 때 도움을 준다. 디렉토리를 찾아 보는 대신 du 명령어는 파일 시스템만을 찾는다.

▶ /home 파일 시스템에 관련한 파일만을 보기

```
# du -x /home
```

1	/home/opmon/.dt/icons
1	/home/opmon/.dt/appmanager
25	/home/opmon/.dt/help/opmon-kpic-0

/ (루트) 파일 시스템을 찾아 보려 한다면, -x 옵션만이 실질적인 방법이다. -x를 사용하지 않으면 시스템의 모든 디렉토리를 보게 된다.

## 파일제거를 위한 명령어

- skulker 명령어는 원하지 않거나 또는 안쓰이는 파일을 제거함으로써 파일 시스템을 정리하는 명령어 이다.
- skulker 명령어로 제거되는 전형적인 파일
  - ▷ /tmp 디렉토리에 있는 파일 : 임시저장 디렉토리
  - ▷ core 파일: 비정상종료된 프로그램에 의해 종료전 프로세서 에서 메모리의 이미지를 떼서 만들어지는 파일
  - ▷ a.out 파일: 소스를 컴파일할 경우 생기는 실행파일
- 일반적으로 skulker는 cron 명령어에 의해 매일 수행된다.

### Notes :

파일을 제거하는 한가지 방법은 /usr/sbin/skulker 명령어를 사용하는 것이다. skulker 명령어는 원하지 않거나 또는 안쓰이는 파일을 제거함으로써 파일 시스템을 정 리하는 명령어이다.

- 제거하기 위한 skulker 파일의 항목은 다음과 같다.
  - ▶ 오래된 모든 qdir 파일
  - ▶ 메일 큐에 남아있는 모든 파일
  - ▶ /tmp 디렉토리에 있으면서 최근 24시간 동안 사용되지 않았거나 수정된 적이 없는 파일과 이보다 더 오래된 파일
  - ▶ /var/tmp 디렉토리에 있는 파일
  - ▶ 45일 이상이 지난 새로운 아이템
  - ▶ 하루 이상이 지난 \*.bak, proof, gally 그리고 ed.hup 파일
  - ▶ 하루 이상이 지난 .putdir 디렉토리에 있는 임의의 파일
- skulker 명령어는 기본적으로 /usr/sbin 디렉토리에 있는 쉘 프로그램이다. 이 프로그램이 부적절하게 편집되어 있는 경우 예상치 못한 결과를 초래할 수 있기 때문에, skulker를 수정하는 경우에는 연습이 필요하다.



## File Systems 검증

### ■ 명령어 사용법 :

```
# fsck [ -p | -y | -n ] file_system
```

### ■ File Systems 점검 단계 :

- ▷ 오류를 대한 저널 로그의 점검
- ▷ i-node, 간접 데이터 블록, 데이터 블록, 그리고 free list의 점검
- ▷ 파일 크기의 점검
- ▷ 디렉토리 항목의 점검

## Notes :

fsck 명령어 : 파일 시스템의 일관성을 검사하고 대화식으로 파일 시스템을 복원

파일 시스템의 무결성을 점검하고 대화식으로 복구하기 위해 fsck 명령어를 사용하는데 이 때, 파일 시스템 이름을 지정하지 않으면, /etc/filesystems 파일에 check=true로 설정되어 있는 모든 파일 시스템을 점검한다. 이 명령은 각 블록이 단일 파일 또는 사용하지 않은 목록에 할당되어 있는지 확인하기 위해 블록을 점검한다.

### ■ 파일 시스템의 일관성을 검사하고 대화식으로 파일 시스템을 복원

```
# fsck [ -n ] [ -p ] [ -y ] [ -V VfsName ] [ FileSystem1-FileSystem2 ... ]
```

-p 옵션 : 자체적으로 데이터 손실이 없도록 최소한으로 복구한다.

-y 옵션 : fsck 명령어의 모든 프롬프트에 yes로 답하도록 한다.

-n 옵션 : fsck 명령의 모든 질문에 대한 응답을 no로 답하도록 한다.

- ▶ 디폴트 파일 시스템(/etc/filesystems)을 사용하여 사소한 문제점을 자동으로 수행

```
# fsck -p
```

## Unit 8. Paging Space

---

Paging Space 란?

메모리 관리

가상메모리 관리자 (VMM)

Paging Space 공간 크기

Paging Space 모니터링

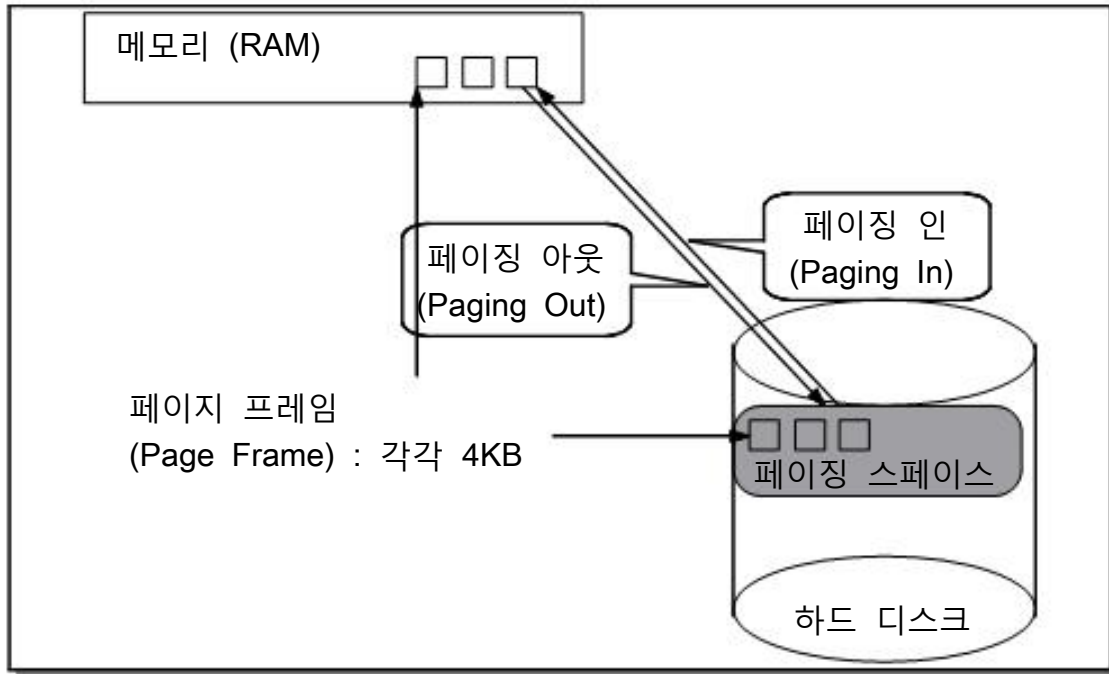
Paging Space 공간 배치

Paging Space 공간 추가

Paging Space 속성 변경

Paging Space 삭제

## Paging Space 란?



### Notes :

#### ■ 페이징 스페이스 (Paging Space)

더 많은 실메모리 또는 임의 액세스 메모리(RAM)를 원하는 경우, 사용자가 원하는 만큼 구매할 수 있다. 그러나, RAM의 가격이 시간이 갈수록 떨어지기는 하지만 추가 RAM을 구매하기에는 여전히 비싸므로 페이징 공간 사용은 RAM을 구매하는데 대한 대안이다.

페이징 공간은 임시 저장 공간으로, 하드 디스크에 위치해 있다. 프로그램의 비활성 부분을 하드 디스크에 저장하면 다른 프로그램을 위해 더 많은 RAM을 남겨두므로 그 만큼의 더 많은 메모리를 사용할 수가 있다. 즉, 이러한 임시 공간을 페이징 스페이스(Paging Space)라 한다.

#### ■ 페이지 프레임 (Page Frame)

RAM 및 페이징 공간은 페이지 프레임이라는 4 KB 섹션으로 나뉘어져 있다. RAM에 있는 모든 페이지 프레임에 대해, 페이징 공간의 하나 이상의 페이지 프레임이 하드 디스크 상에 있다. 시스템이 더 많은 RAM을 필요로 할 때, 정보의 페이지 프레임은 RAM에서 나와 하드 디스크로 이동한다. 이를 페이지 아웃(paging out)이라고 한다 이들 정보 페이지 프레임이 다시 필요해지면, 하드 디스크에서 가져와 다시 RAM으로 이동시킨다. 이를 페이지 인(paging in)이라고 한다.

## 메모리 관리

### ▣ 실 메모리의 보유량 보기

```
# lsattr -El sys0 -a realmem
```

```
realmem 262144 사용 가능한 물리적 메모리의 크기(KB)
```

```
: 현재 총 메모리는 262144KB입니다. 위의 메모리를 명령어로 변경 할 수 없습니다.
```

### Notes :

사용자 컴퓨터에 있는 실메모리의 양은 사용자의 옷장 공간과 같다. 현재 보유하고 있는 정도에 관계 없이 사용자는 언제나 보다 많이 필요로 한다.

실메모리 보유량을 확인하려면, 리스트 속성 명령을 사용하여야 한다.

```
# lsattr -El sys0 -a realmem
```

#### ■ 시스템의 장치에 대한 속성 특성 및 가능한 속성값을 표시.

```
# lsattr[ -E ] -l Name [ -a Attribute ]
```

-E 옵션 : 유효한 값을 표시.

-l 옵션 : 장치 논리적 이름을 속성 이름 또는 값이 표시될 사용자 조정된 장치 오브젝트 클래스에 지정.

-a 옵션 : 특정 장치나 장치의 종류에 지정된 속성에 대한 정보를 표시.

#### ▶ 테이프 장치 rmt0에 대한 현재 속성 값을 나열하기

```
# lsattr -l rmt0 -E 3962
```

## Paging Space 모니터링

▣ 모든 페이징 공간 나열

# lsps -a

페이지공간	물리적볼륨	볼륨그룹	크기	사용률%	활동중	자동	유형
hd6	hdisk0	rootvg	2048MB	41	yes	yes	lv

▣ 모든 페이징 공간을 요약해서 나열

# lsps -s

총 페이징 공간	사용률 %
2048 MB	41

### Notes :

위의 그림은, 페이징 공간 hd6는 볼륨 그룹 rootvg에 있는 하드디스크 hdisk0에 위치하고 있으며, 크기는 2048MB이고 이 2048MB의 41%가 현재 사용 중이라는 설명.

- ▶ 활동 중 yes 값 : 페이징 공간 hd6가 현재 사용 중임을 나타낸다.
- ▶ 자동 yes 값 : hd6가 시스템이 시동될 때마다 자동으로 시작된다는 것을 나타낸다.
- ▶ lv 값 : hd6가 논리 볼륨임을 의미한다

#### ■ 페이징 공간의 특성 표시

# lsps { -s | -a } PagingSpace

-a 옵션 : 모든 페이징공간의 특성 표시

-s 옵션 : 모든 페이징공간의 특성을 요약해서 출력

- ▶ 모든 페이징 공간의 특성을 나열하기

# lsps -a

충분한 페이징 공간이 있는지 확인하려면, 사용률% 값을 보아야 한다. 이 값은 사용된 페이징 공간 백분율이 30%에서 70% 사이에 있음을 나타낸다.

이 백분율은 몇 명의 사용자가 시스템에 있으며 어느 프로그램이 실행 중인지에 따라 계속 변한다. 사용자는 시스템 관리자로서 주간 내내 하루에 여러 번 페이징 공간 사용 백분율을 모니터해야 한다. 최대 사용자 시간을 판단하려면 시스템 사용 추세를 살펴보아야 한다.

사용률%가 계속해서 70%를 초과하면, 더 많은 페이징 공간이나 메모리를 시스템에 추가해야 한다.

페이징 공간에서 시스템이 느리게 실행할 경우, 메시지를 콘솔에 보내고 때로는 사용자에게도 보낸다. 이 때, 시스템은 일부 실행 중인 프로세스가 종결되거나 일부 할당된 메모리가 릴리스 될 때까지 새로운 프로세스를 시작할 수 없다. 이러한 상황은 피해야 한다.

다음과 같은 메시지가 콘솔에 표시되거나 명령에 대한 응답으로 단말기에 표시되는 것은 페이징 공간 부족을 나타낸다.

- ▶ INIT 페이징 공간이 부족함
- ▶ 메모리가 충분하지 않음
- ▶ Fork 기능 실패
- ▶ Fork() 시스템 호출 실패
- ▶ Fork 할 수 없습니다. 프로세스가 너무 많습니다.
- ▶ Fork 실패 사용할 수 있는 메모리가 충분하지 않음
- ▶ Fork 기능을 지원하지 않음. 사용할 수 있는 메모리가 충분하지 않음
- ▶ Fork 할 수 없음. 공간이 충분하지 않음

이러한 상황은 악화되므로 사용자는 충분한 페이징 공간을 가지고 있는지 확인을 해야 한다.

페이징 공간이 가득 찰 경우, 시스템 외의 프로세스가 종결되고 시스템이 정지 할 수도 있다.

## Paging Space 공간 추가

# smit mkps

다른 페이징 공간 추가

입력 필드에 값을 입력하거나 선택하십시오. 원하는 것을 모두 변경한 후, Enter 키를 누르십시오.

	[입력 필드]	
볼륨 그룹 이름	rootvg	
페이징 공간 크기(논리적 계획)	1	#
물리적 볼륨 이름		+
이 페이징 공간을 지금 시작합니까?	아니오	+
시스템을 다시 시작할 때마다 이 페이징 공간을 사용합니까?	아니오	+

F1=도움말

F5=재설정

F9=셸

F2=화면갱신

F6=명령

F10=종료

F3=취소

F7=편집

Enter=실행

F4=리스트

F8=이미지

### Notes :

smit mkps 명령어 : 페이징공간을 만들기 위한 smit fast path

■ 페이징 공간의 크기(SIZE of paging space) 항목 :

입력 필드에 페이징 공간의 크기를 표시한다.

바람직한 것은 동일한 크기 또는 거의 동일한 크기를 갖는 여러 개의 페이징 공간을 각기 다른 하드 디스크에 두는 것이다.

■ 물리적 볼륨이름(PHYSICAL VOLUME name) 항목 :

페이징 공간을 위해 어느 하드 디스크를 사용할 지를 표시하기 위해 입력 항목을 사용한다.

■ 페이징공간을 지금 시작합니까?(Start using this paging space NOW?) 항목 :

AIX가 새로운 페이징 공간을 즉시 사용할 지 또는 사용하지 않으면서 생성만 할 지를 결정한다.

■ 추가 페이징 공간을 시스템에 추가

# mkps [ -a ] [ -n ] -s LogicalPartitions VolumeGroup

-a 옵션 : 페이징 공간이 다음 재시작 시에 구성 설정됨을 지정

-n 옵션 : 페이징 공간을 즉시 활성화

-s 옵션 : LogicalPartitions 논리적 구획에 만들어질 페이징 공간과 논리적 볼륨의 크기를 지정

- ▶ myvg 볼륨 그룹에서, 네개의 논리적 구획을 갖고, 후속의 모든 시스템 재시작 시에 즉시 활성화되는 페이징 공간을 작성하기

# mkps -a -n -s4 myvg



## Paging Space 삭제

# smit rmps

페이징 공간 제거

입력 필드에 값을 입력하거나 선택하십시오. 원하는 것을 모두 변경한 후, Enter 키를 누르십시오.

페이징 공간 이름

[입력 필드]

+

F1=도움말	F2=화면갱신	F3=취소	F4=리스트
F5=재설정	F6=명령	F7=편집	F8=이미지
F9=셸	F10=종료	Enter=실행	

### Notes :

smit rmps : 페이징공간을 삭제하기 위한 smit fast path

페이징 공간 제거(rmps) 명령을 사용하여 비활성 페이징 공간을 삭제할 수 있다. 예를 들어, paging01을 제거하려면, smit rmps paging01을 입력한다. 이 명령을 입력하면, 페이징 공간 제거 메뉴가 화면과 같이 표시 된다.

/dev/hd6 페이징 공간은 시스템이 부트 할 때 필요하기 때문에 비활성화 및 제거가 불가능하다.

■ 페이징 공간이 있는 논리적 볼륨과 함께 시스템에서 페이징 공간을 제거  
# rmps PagingSpace

▶ PS01 페이징 공간을 제거하기  
# rmps PS01

## Unit 10. AIX Software 설치 및 유지보수

---

Software 설치 및 유지보수

설치된 Software 목록

Software 유지보수와 유틸리티

instfix 명령어

# Software 설치 및 유지보수

```
# smit install
```

## 소프트웨어 설치 및 유지보수

원하는 항목으로 커서를 옮긴 후, Enter 키를 누르십시오.

- 소프트웨어 설치 및 갱신
- 소프트웨어 및 관련 정보 리스트
- 소프트웨어 유지보수 및 유틸리티
- 네트워크 설치 관리
- 시스템 백업 관리 프로그램

F1=도움말

F2=화면갱신

F3=취소

F8=0 이미지

F9=월

F10=종료

Enter=실행

### Notes :

## ■ Install and Update Software

소프트웨어를 설치 또는 갱신

## ■ List Software and Related Information 소프트

웨어의 목록과 관련 정보의 목록을 출력

## ■ Software Maintenance and Utilities

소프트웨어의 유지보수 및 유틸리티

## ■ Network Installation Management

## 네트워크의 설치 및 관리

■ System Backup Manager

시스템 백업 관리 프로그램

## 설치된 Software 목록

# lspp -l bos.\*

파일세트	레벨	상태	설명
경로: /usr/lib/objrepos			
bos.64bit	4.3.3.76	확정	Base Operating System 64 bit Runtime
bos.acct	4.3.3.79	확정	Accounting Services
bos.adt.base	4.3.3.77	확정	Base Application Development Toolkit
bos.adt.debug	4.3.3.78	확정	Base Application Development Debuggers
bos.adt.graphics	4.3.3.75	확정	Base Application Development Graphics Include Files
bos.adt.include	4.3.3.81	확정	Base Application Development Include Files
bos.adt.lib	4.3.3.10	확정	Base Application Development Libraries
bos.adt.libm	4.3.3.50	확정	Base Application Development Math Library
bos.adt.prof	4.3.3.81	확정	Base Profiling Support

### Notes :

■ 소프트웨어 제품을 나열

# lspp { -l | -p } [ -q ] [ FilesetName ... | FixID ... | all ]

-l 옵션 : 지정된 파일세트의 이름, 가장 최신 레벨, 상태 및 설명을 표시

-p 옵션 : 지정된 파일세트에 대한 필수 정보를 표시

-w 옵션 : 이 파일을 소유한 파일세트를 나열합니다. 이 플래그는 AIX 버전 4.2 이상 버전에 적용

## Software 유지보수와 유틸리티

### # smit maintain\_software

#### 소프트웨어 유지보수 및 유틸리티

원하는 항목으로 커서를 옮긴 후, Enter 키를 누르십시오.

적용된 소프트웨어 갱신사항 확정 (보관된 파일 제거)  
적용된 소프트웨어 갱신사항 거부 (이전 버전 사용)  
설치된 소프트웨어 제거

나중 설치를 위해 하드 디스크에 소프트웨어 복사

설치 후 소프트웨어 파일 크기 점검  
소프트웨어 설치 필수 요건 검증

실패 또는 인터럽트된 설치를 원상복구합니까?

F1=도움말

F2=화면갱신

F3=취소

F8=이미지

F9=셸

F10=종료

Enter=실행

### Notes :

- **Commit Applied Software Updates (Remove Saved Files)**  
적용된 소프트웨어의 갱신사항을 확정한다. 보관된 파일(이전 버전)은 제거된다.
- **Reject Applied software Updates (Use Previous Version)**  
적용된 소프트웨어의 갱신사항을 거부한다. 이전에 보관되었던 소프트웨어 버전을 다시 사용한다.
- **Remove Installed Software**  
설치된 소프트웨어를 제거한다.
- **Copy Software to Hard Disk for Future Installation**  
나중에 설치가 가능하도록 하드디스크에 소프트웨어를 복사하여 보관한다.
- **Check Software File Sizes After Installation**  
설치 후 소프트웨어의 파일 크기를 점검 한다.
- **Verify Software Installation and Requisites**  
소프트웨어를 설치함에 있어서 필수 요건을 검증한다.
- **Clean Up After Failed or Interrupted Installation**  
실패 또는 인터럽트가 된 소프트웨어 설치 과정을 원상 복구한다.

## instfix 명령어

### ■ fix 설치

```
# instfix -k IX38794 -d /dev/rmt0.1
```

### ■ fix 검색

```
# instfix -ik IX38794
All filesets for IX38794 were found.
```

### ■ 키워드를 이용한 fix 검색

```
# instfix -s SCSI -d /dev/rmt0.1
```

## Notes :

instfix 명령을 사용하면, 허가된 프로그램 분석 보고서(APAR) 번호나 수정사항을 식별하는 그 밖의 고유 키워드 이외의 정보를 모를 경우에도 수정사항 또는 수정 세트를 설치할 수 있다.

그 밖에도 instfix 명령은 수정사항이 시스템에 설치되어 있는지 여부를 판별하는데도 사용할 수 있다.

### ■ 키워드 또는 수정과 연관된 파일세트를 설치

```
# instfix [ -i | -s String ] [ -k Keyword ] [ -d Device ]
```

-s 옵션 : String 지정된 문자열을 포함하는 미디어에서 수정사항을 검색하고 표시

-k 옵션 : 설치될 APAR 번호 또는 keyword 를 지정하며 여러 개의 키워드를 입력. 입력된 키워드의 리스트는 인용 부호로 묶고 공백으로 분리

-d 옵션 : Device 입력 장치를 지정

-i 옵션 : 수정사항 또는 키워드가 설치되어 있는지의 여부를 표시

- ▶ /dev/rmt0.1에 마운트된 테이프에서 수정사항 IX38794와 연관된 모든 파일세트를 설치하기

```
# instfix -k IX38794 -d /dev/rmt0.1
```

## Unit 11. 백업과 복구

---

백업을 하는 이유  
백업의 종류  
백업 장치  
증진백업 예시  
tar 명령어  
Tape 조작방법  
좋은 습관

## 백업을 하는 이유

---

- 중요한 데이터의 보관을 위해 필요하다.
  - 재해 시 복구를 위해 필요하다.
  - 파일 시스템의 재구성을 위해 필요하다.
  - 시스템 간의 데이터 전송을 위해 필요하다.
  - 설치에 대한 시스템 이미지를 확보하기 위해 필요하다.
  - 버전 비교를 위한 체크포인트로써 필요하다.
  - 장기간 특정 분야에 대한 보관을 위해 필요하다.
- 

### Notes :

컴퓨터에 저장되어 있는 데이터는 일반적으로 매우 중요하며, 대체하려면 기계 자체보다 비용이 더 소모된다. 백업 정책을 가지고 있지 않은 기업은 데이터가 유실되었거나 컴퓨터가 손상되어 데이터를 복구할 수 없게 된 경우, 일반적으로 업무가 중단된다.

백업은 시스템 복구만을 위한 것이 아니다. 파일 시스템이 너무 커진 경우, 이를 줄이는 유일한 방법은 백업을 작성하고 파일 시스템을 제거한 후, 작은 크기로 다시 작성하여, 필요할 때 백업에서 복원시키는 것이다.

백업의 사용은 시스템들이 서로 연결되어 있지 않은 경우 한 시스템에서 다른 시스템으로 파일을 복사하는 유일한 방법이 될 수도 있다.

운영 체제가 파손되었거나 손상되었을 때, 시스템에 있는 데이터를 복구해야 하는 경우, 사용자는 완전한 시스템 백업이 필요하다.



## 백업의 종류

### ■ System Backup(시스템 백업)

- ▷ 논리 볼륨 내부/외부(inter/intra) 할당 정책
- ▷ 스트리핑(striping)
- ▷ 파일 시스템 설정(nbpi, 압축, fragment 크기)
- ▷ 루트 볼륨 그룹만의 페이징 설정을 포함하여
- ▷ 모든 시스템 데이터를 기록

### ■ Full Backup(전체 백업)

모든 사용자 데이터 및 구성 파일을 보존

### ■ Incremental Backup(점증적 백업)

마지막 전체 백업 이후의 변경사항을 기록

## Notes :

시스템 백업은 시스템 파일에 작성된 변경사항을 백업하기 위해 수행된다. 시스템 백업은 시스템이 업그레이드될 때마다 수행되어야 하며 이후에는 스케줄에 따라 수행되어야 한다.

시스템 백업은 root 볼륨 그룹만 보존한다.

### ■ 루트 볼륨 그룹의 설치 가능한 이미지를 파일이나 부트 가능한 테이프에 작성

```
# mksysb [ -b Number ] [ -i | -m ] Device | File
```

-b 옵션 : Number 단일 출력 조작에 기록할 512 바이트 블록 수를 지정

-i 옵션 : mkszfile 명령을 호출하여, /image.data 파일을 생성

-m 옵션 : -m 플래그로 mkszfile 명령을 호출하여 맵 파일을 생성

- ▶ /dev/rmt0이라는 이름의 테이프 장치에 시스템 백업을 생성하고, /image.data 파일(mkszfile 명령으로 생성)을 작성하기

```
# mksysb -i /dev/rmt0
```

## 증진백업의 예시

Sun	Mon	Tue	Wed	Thur	Fri	Sat
					1 level 0	2
3	4 level 6	5 level 6	6 level 6	7 level 6	8 level 3	9
10	11 level 6	12 level 6	13 level 6	14 level 6	15 level 0	16
17	18 level 6	19 level 6	20 level 6	21 level 6	22 level 3	23
24	25 level 6	26 level 6	27 level 6	28 level 6	29 level 0	30
31						

### Notes :

첫번째 월요일에 수행되는 레벨 6 백업은 레벨 0 백업이 지난 금요일에 수행된 이후 수정된 모든 파일을 백업한다.

화요일부터 목요일까지의 각 레벨 6 백업은 하위 레벨 백업(이 경우, 레벨 0) 이후 수정된 모든 파일을 백업한다.

레벨 3 백업은 이전 레벨 0 백업 이후 수정된 모든 파일을 백업한다. 두 번째 전체 주의 레벨 6 백업은 이제 레벨 3인 하위 레벨 백업 이후 수정된 모든 파일을 백업한다.

앞의 예제에서는 번호 3과 6이 임의적으로 선택되었다.

순서없이 번호를 사용하면 필요한 경우 백업 간에 다른 점증을 수행할 수 있다.

레벨 6 점증적 백업은 월요일부터 목요일까지 수행되므로, 백업의 크기는 변경된 사용자 데이터의 양에 따라 증가한다.

## tar 명령어

### ■ tar 백업 생성

```
# tar -cvf /dev/rmt3 /home
```

### ■ tar 이미지로부터의 파일 복원

```
# tar -xvf /dev/rmt0 /home/team01/mydir
```

### ■ tar 파일의 내용 출력

```
# tar -tvf /dev/rmt0
```

## Notes :

tar(tape archive) 명령은 거의 모든 유형의 UNIX 플랫폼에서 사용하는 일반적인 명령이다. tar 명령은 단일파일이나 디렉토리 또는 tar 이미지의 전체내용을 복원하는 데 사용이 가능하다. 루트 사용자가 이미지를 복원하면, 원래 소유권과 권한이 보존된다. 루트 이외의 사용자가 tar 이미지를 복원하면, 그 사용자가 모든 파일을 소유한다.

### ■ 테이프 또는 디스켓이나 tar 파일을 생성

- ▶ /home에서 시작하는 모든 파일과 디렉토리를 포함할 경우  
# tar [-옵션] [저장장치] [인수]

- c 옵션 : 새 아카이브를 만듦
- v 옵션 : 파일을 처리할 때마다 파일 이름을 보여줌
- f 옵션 : 변수를 읽거나 쓸 아카이브 매체를 지정

```
# tar -cvf /dev/rmt0.3 /home
```

- ▶ /home/test01/mydir과 이 디렉토리의 모든 내용을 복원할 경우  
-x 옵션 : 아카이브 사본에서 하나 이상의 파일 매개변수로 지정한 파일들을 발췌

```
# tar -xvf /dev/rmt0 /home/test01/mydir
```

- ▶ tar 이미지의 내용을 읽고 파일 및 디렉토리 이름을 나열할 경우  
-t 옵션 : 아카이브에 나타나는 순서로 파일을 나열

```
# tar -tvf /dev/rmt0
```

## Tape 조작방법

# tctl

- rewind      - rewinds a tape
- fsf          - fast forwards a tape
- offline      - ejects a tape
- rewoffl      - rewinds and ejects a tape

```
# tctl -f /dev/rmt0 rewind
# tctl -f /dev/rmt0.1 fsf 3
# tctl -f /dev/rmt0 rewoffl
```

# restore -s

```
# restore -s 4 -xvf /dev/rmt0.1 ./etc/inittab
```

### Notes :

- fsf 하부 명령은 테이프를 지정한 파일 개수 표시만큼 앞으로 이동한다.
  - ▶ 테이프의 위치를 세 번째 파일에서 파일 두 개 만큼 앞으로 옮기는 경우
 

```
# tctl -f /dev/rmt0.1 fsf 2
```
- bsf 하부 명령은 테이프를 지정한 파일 표시 수 만큼 뒤로 이동 시킨다.
  - ▶ 테이프를 현재 위치에서 한 개의 파일 뒤로 이동 시키는 경우
 

```
# tctl -f /dev/rmt0.1 bfs 1
```
- rewind 하부 명령은 테이프 되감기를 한다.
 

```
# tctl -f /dev/rmt0 rewind
```

## 좋은 습관

---

- ▣ 백업을 항상 검증한다. (Verify your backups)
- ▣ Tape 장치를 체크한다. (Check the tape device)
- ▣ 이전 백업을 보관한다. (Keep old backups)
- ▣ 떨어져 있는 안전한 장소에 보관한다. (Offsite, secure storage)
- ▣ Tape에 이름표를 부착한다. (Label tape)
- ▣ 미리 복구절차를 테스트 해 본다. (Test recovery procedures before you have to!)

---

### Notes :

다음은 사용자의 백업 정책이 효과적인지를 확인하기 위해 수행할 수 있는 단계이다.

1. fsck 명령을 사용하여 백업하기 전에 파일 시스템을 점검한다.
2. fuser 명령을 사용하여 백업 중 사용되고 있는 파일이 없는지 확인한다.
3. 사용된 명령 및 블록 크기를 포함하여 백업 매체가 명백하게 레이블 되어 있는지 확인한다.
4. 사용된 테이프 번호 또는 기타 매체 ID를 포함하여 백업된 파일의 하드카피 리스팅을 보존한다.
5. tapechk 명령을 주기적으로 사용하여 백업을 점검한다.
6. 안전한 오프사이트 기억장치에 백업 매체를 보존한다.
7. 이전 백업을 보존한다.
8. 이들을 사용하기 전에 복구 절차를 테스트한다.

## Unit 12. 보안과 사용자 관리

---

root 접근 관리

보안 Log

보안과 사용자

root의 password 되찾기

SMIT 그룹

---

## root 접근 관리

---

- ▣ 권한이 있는 로그인 액세스를 제한한다.
- ▣ 주기적으로 루트 사용자의 암호를 변경한다.
- ▣ 다른 시스템에는 다른 루트 암호를 설정.
- ▣ 시스템 관리자는 항상 일반 사용자 계정으로 로그인 한 다음, su 명령어를 사용하여 루트 계정으로 사용자를 전환해야 한다.
- ▣ 루트 사용자가 사용하는 PATH는 노출되어서는 안된다.

---

### Notes :

유닉스/AIX 세계에서 모든 사람이 알고 있는 유일한 사용자 계정은 루트이다. 그러므로 임의의 사용자에게 의한 시스템 액세스를 중지하는 것은 루트 암호 하나 뿐이다.

su(switch user) 명령은 sulog 파일에 의해 감사 추적을 제공한다.

루트 사용자가 사용하는 PATH는 노출되어서는 안되므로, PATH 환경변수 설정 시 루트 사용자에게 의해 단독으로 관리되는 임의의 디렉토리를 포함하지 않도록 해야한다.

다른 사용자가 루트 사용자의 PATH 항목에 있는 임의의 디렉토리에 파일을 추가할 수 있는 권한이 있을 경우, 시스템을 손상시킬 수 있는 잘못된 파일을 실수로 실행할 수도 있다

## 보안 Log

/var/adm/sulog



su를 사용한 흔적을 감시

/var/adm/wtmp



성공적인 로그인 정보

/etc/utmp



일반 user의 로그인 정보 출력

/etc/sevurity/failedlogin



실패한 로그인 정보

## Notes :

루트 액세스를 제어하는 방법 중 하나는 시스템 관리자가 su(switch user) 명령을 사용하는 것이다. su 명령은 루트나 다른 사용자로 전환하기 위해 사용할 수 있다. 명령에 - 를 명시하면, 환경은 루트 또는 명시된 사용자로 변경된다.

예를 들어, 아래의 명령은 환경을 사용자가 test01로 로그인한 것으로 설정한다.

```
# su - test01
```

사용자가 시스템을 액세스할 때 마다, 사용자에 대한 정보가 한 개 이상의 파일에 기록 된다. 누가 시스템을 액세스 했고 또는 액세스를 시도하였는지 보기 위해 주기적으로 이러한 파일들을 검토하는 것이 좋다.

su 명령이 실행되면, 아래의 내용이 /var/adm/sulog 파일에 기록된다.

- ▶ su 명령을 실행한 날짜와 시간
- ▶ 액세스의 성공 여부
- ▶ 명령이 실행된 터미널
- ▶ 그 명령을 실행한 계정
- ▶ 사용자가 변경을 시도한 계정



권한이 없는 사용자에 의한 다수의 시도는 사용자 계정을 이용해서 누군가가 시스템에 침입하려 했음을 나타낸다.

사용자가 로그인에 성공하면, 로그인 프로그램은 모든 로그인의 기록인 `/var/adm/wtmp` 파일과 현재 로그인되어 있는 사용자의 목록인 `/etc/utmp`에 기록한다.

이들 파일을 보려면 `who` 명령을 사용한다. 기본으로 `who` 명령은 `/etc/utmp`를 찾는다. 다른 파일을 보기 위해서는 `who /var/adm/wtmp`를 입력한다.

`/etc/security/failedlogin` 파일에서 시스템에 로그인 하기 위한 반복적인 시도를 발견할 수 있다.

아래와 같이 `who` 명령을 사용하여 이 파일을 볼 수 있다.

```
# who /etc/security/failedlogin
```

이 파일의 내용에 대한 정기적 점검에 추가하여, 그 크기를 모니터링하고 더 나아가 갑자기 크기가 증가하는지도 살펴보는 것이 좋다.

## 보안과 사용자

### # smit security

보안 및 사용자			
원하는 항목으로 커서를 옮긴 후, Enter 키를 누르십시오.			
사용자 그룹 암호 로그인 제어 역할			
F1=도움말	F2=화면갱신	F3=취소	F8=이미지
F9=첼	F10=종료	Enter=실행	

### Notes :

- Users : 시스템에 사용자를 추가하고 기존 사용자를 삭제하며, 기존 사용자의 속성을 변경하는데 사용된다.
- Groups : 시스템에 그룹을 추가하고 그룹을 삭제하며, 기존 그룹의 속성을 변경하는데 사용된다.
- Passwords : 사용자에게 대한 암호를 변경하는데 사용된다. 이것은 새로운 사용자를 설정하거나 사용자가 암호를 잊어버린 경우에 필요하다
- Login Controls : 사용자 계정 및 특정 단말기에 제한된 액세스 기능을 제공한다.
- Roles : 시스템에 있는 기존 역할을 보고 작업하며 새로운 역할을 생성하기 위한 메뉴를 제공한다. 사용자가 루트 권한을 필요로 하는 기능을 실행할 수 있도록 하는 0 이상의 인증으로 구성되어 있다.

## root의 Password 되찾기

---

- CD-ROM 또는 부팅이 가능한 Tape 으로 부팅한다.
  - Installation and Maintenance 메뉴에서 3번 옵션, Maintenance Mode for System Recovery를 선택한다.
  - 루트 볼륨 그룹을 활성화하여 shell을 여는 옵션을 따른다.
  - TERM 변수를 설정하고 export 한다.
  - vi /etc/security/passwd 명령어를 입력 후 루트의 암호화된 암호를 삭제한다.
  - sync; sync 명령어를 입력한다.
  - 시스템을 재시작 한다.
  - 루트 암호를 설정한다.
- 

### Notes :

사람들은 종종 자신의 암호를 잊어 버린다.

만약 시스템 관리자가 루트 암호를 잊는 경우, 다시 회복해야 한다.  
위의 단계로 암호를 회복할 수 있다.

# SMIT 그룹

```
# smit groups
```

그룹

원하는 항목으로 커서를 옮긴 후, Enter 키를 누르십시오.

모든 그룹 나열

## 그룹 추가

그룹의 특성 변경 / 표시

## 그룹 제거

F1=도움말

F2=화면갱신

F3=취소

F8=0|□|ㄱ|

F9=셀

F10=종료

Enter=실행

### Notes :

그룹의 목적은 여러 사용자가 한 개 이상의 파일에 대해 완전히 동일한 권한을 갖도록 하는 데 있다.

시스템에는 이미 몇 개의 그룹이 있다. 여기에는 루트가 속한 시스템(system) 그룹과 일반 사용자가 속한 스태프(staff) 그룹이 포함되어 있다.

그 외 그룹은 몇몇 응용 프로그램과 파일을 위해 존재한다. 예를 들어 누군가 회계 (accounting) 또는 급여 (payroll) 그룹의 일부가 될 수 있다.

그룹을 생성하고 해당 그룹의 관리자가 되려면 루트 사용자 또는 보안(security) 그룹의 구성원이어야 한다.

List All Groups 메뉴를 선택하면 그룹 목록이 표시된다. 이 목록을 표시할 수 있는 또 다른 방법은 lsgroup ALL 명령어를 사용하는 것이다.

전체 명령어 형식은 다음과 같다.

```
# lsgroup [-c-f][-a attribute](ALL[groupname])
```

Add a Group 메뉴에서 ADMINISTRATOR list 는 USER list에서 그룹의 특성을 변경하고, 구성원을 추가 또는 삭제할 수 있는 구성원의 목록이다.

참고로 사용자는 최대 32개 그룹까지 속할 수 있다.

Change Group Attributes 메뉴에서 그룹 정보를 변경할 수 있다.

Remove a Group 메뉴에서는 그룹을 삭제 할 수 있다. 단, 루트 사용자만이 관리자 그룹을 삭제할 수 있다. 하지만, 그룹이 누군가의 1차 그룹인 경우, 그 그룹은 삭제할 수 없다.

## Unit 13. Networking Overview

---

TCP/IP는 무엇인가

TCP/IP 설정에 필요한 정보

TCP/IP 설정

Name 분석

Hostname 확인

# TCP/IP는 무엇인가

- Transmission Control Protocol / Internet Protocol
- 상이한 네트워크 상의 서로 다른 시스템간 data를 교환하기 위한 소프트웨어
- 시스템이 사용자간에 투명성 있게 연결하고 data를 보내는지에 대한 방법
- TCP/IP는 vendor에 독립적

## Notes :

TCP/IP는 data를 교환할 어떤 네트워크의 종류에 의해 연결된 컴퓨터간의 협동을 위한 방법을 정의한 네트워킹 구조이다. TCP/IP는 메인프레임에서 일반 개인용 컴퓨터까지의 모든 platform을 모두 커버할 수 있는 방법을 가지고 있는 소프트웨어다.

TCP/IP는 네트워크 상에서 서로 다른 두 컴퓨터가 연결하기 위한 여러 가지 방법을 정의한 프로토콜이다.

TCP/IP는 어떤 시스템 장비, 네트워크 하드웨어, 운영체제에나 독립적인 프로토콜이다. 이것은 서로 다른 vendor의 장비나, 다른 타입의 네트워크, 서로 다른 운영 환경의 장비 사이에서도 연결이 만들어질 수 있다는 것을 의미한다.

TCP/IP는 미 국방성이 이기종간의 컴퓨터간 적은 비용으로 통신을 하기 위한 프로토콜이 필요하다는 요구에 의하여 1968년 DARPA(Defense Advanced Research Projects Agency)의 지원으로 만들어진 ARPANET가 발전된 것이다.

## TCP/IP 설정에 필요한 정보

### ■ Address

- ▷ 각 adapter는 유일한 TCP/IP주소와 subnet mask를 가진다.  
그리고 그것들은 관리자로 하여금 할당된다.

### ■ Name

- ▷ 각 장비는 유일한 hostname을 가진다.
- ▷ 주소 해석을 위한 name table를 가지고 있어야 한다.

### ■ Routes

- ▷ 다른 네트워크의 시스템과 통신을 위해 gateway 주소가 필요하다.

## Notes :

TCP/IP 네트워크에 있는 각 시스템은 반드시 유일한 TCP/IP 주소와 hostname을 가지고 있어야만 한다. 따라서 네트워크 관리자는 주소와 name table를 중심으로 시스템에 IP주소와 hostname을 할당한다.

AIX V4.3은 IPV4와 IPV6 주소를 모두 지원한다. IPV6 주소는 128bits의 길이를 가지며 콜론으로 구분된 16-bit의 8개 필드로 표시한다. IPV4와 IPV6를 같이 사용하기 위해 터널링이라는 기술이 사용된다.

네트워크의 각 시스템은 연결을 위해 hostname을 찾고 기억한다. 그러나 TCP/IP 프로토콜은 data를 보낼 때 TCP/IP 주소만을 사용한다. 그러므로 TCP/IP는 hostname과 TCP/IP주소를 번역해주는 기능을 한다. 이것을 name resolution이라고 한다.

### ■ Name resolution을 하는 방법

#### ▶ Flat Network :

네트워크의 각 호스트는 연결하기 위한 호스트에 관한 이름과 주소를 /etc/hosts 라는 파일에 가지고 있다. 이 방법은 간단하면서도 빠른 방법이지만 호스트가 너무 많은 경우는 관리에 어려움이 있다.

#### ▶ Domain Network (Domain Name System) :

호스트는 파일 디렉토리와 비슷한 구조의 계층 구조인 도메인 그룹을 가지고 도메인(네임서버)의 하나 또는 그 이상의 호스트는 모든 호스트의 이름과 주소를 가지고 있다. client 호스트는 /etc/resolv.conf 파일에 적힌 네임서버에 연결하기 위한 호스트의 이름과 주소 정보를 요청한다.



## TCP/IP 설정

# smit mktcpip

최소 구성 및 시작			
기존 구성설정 자료를 삭제하려면, 자세한 구성설정 메뉴를 사용하십시오.			
입력 필드에 값을 입력하거나 선택하십시오.			
원하는 것을 모두 변경한 후, Enter 키를 누르십시오.			
* 호스트 이름	[입력 필드]		
	[kpic]		
* 인터넷 주소(점분리 십진수)	[211.196.145.105]		
네트워크 마스크(점분리 십진수)	[255.255.255.192]		
* 네트워크 인터페이스	en0		
네임 서버			
인터넷 주소(점분리 십진수)	[168.126.63.1]		
도메인 이름	[.]		
디폴트 게이트웨이 주소	[211.196.145.65]		
(점분리 십진수 또는 상징적 이름)			
사용자의 케이블 유형	N/A		+
즉시 시작	아니오		+
F1=도움말	F2=화면갱신	F3=취소	F4=리스트
F5=재설정	F6=명령	F7=편집	F8=이미지
F9=셸	F10=종료	Enter=실행	

### Notes :

- 호스트 이름 : 시스템에 사용할 이름
- 인터넷 주소 : 시스템이 사용할 인터넷 주소로 관리자에 의해 지정되는 유일한 식별 값
- 네트워크 마스크 : 네트워크 구분을 위해 사용되는 주소로 관리자에 의해 지정된다.
- 네트워크 인터페이스 : 시스템에서 사용되는 네트워크 카드
- 네임 서버 : name resolution을 위해 사용할 서버의 주소
- 도메인 이름 : 네임 서버의 이름
- 디폴트 게이트웨이 주소 : 다른 네트워크와 연결되어 있는 시스템의 주소로 자신이 속한 네트워크 뿐만 아니라 다른 네트워크를 통한 인터넷을 사용하기 위해서는 반드시 알아야만 한다.
- 사용자의 케이블 유형 : 시스템이 사용하는 케이블 유형을 선택하는 것으로 bnc(동축 케이블, dix(이더넷 케이블), tpo(utp같은 쌍으로 꼬인 케이블)을 선택한다.
- 즉시 시작 : 설정을 마친 후 네트워크를 바로 시작할지 다음 부팅 후 시작할지 선택

## Name 분석

# more /etc/hosts

```
#The format of this file is:
#Internet Address Hostname #Comments

#Items are separated by any number of blanks or tabs. A "#" indicates the
beginning of a comment; characters up to the end of the line are not
interpreted by routines which search this file. Blank lines are allowed in this
file.

#Internet Address      Hostname      #Comments
127.0.0.1              loopback     localhost
192.9.200.1            sys1         timeserver
192.9.200.2            sys2
192.9.200.3            sys3
192.9.200.4            sys4
```

## Notes :

호스트 name과 internet 주소는 /etc/hosts 파일의 값들에 의해 mapping 된다. 일반적으로 네임서버가 down이 되었을 경우 /etc/hosts 파일의 값들을 이용하게 된다. AIX 사용자는 SMIT나 vi 명령을 이용해 /etc/hosts 파일에 값을 추가할 수 있다.

## Hostname 확인

### ■ hostname Command

```
# hostname
sys3
```

### ■ host Command

```
#host sysd3
sys3 is 9.25.35.10, Aliases: sys3.washington.ibm.com

#host 9.25.35.10
sys3 is 9.25.35.10, Aliases: sys3.washington.ibm.com
```

## Notes :

- hostname : 장비의 이름을 확인 하는데 사용되는 명령어
- host : 호스트 이름과 함께 사용될 경우에는 IP 주소를 확인하고 IP 주소와 함께 사용될 경우에는 호스트 이름을 확인하는데 사용되는 명령어