EnterpriseSMS Disk Management (UNIX)

Concept

UNIX provides a number of advanced utilities to manage the physical disk drives attached to a given system. Physical disk drives are defined as *physical volumes* when installed. Physical volumes are then assigned to *volume groups*, permitting them to be managed as a single unit. Each volume group may include up to 32 physical volumes. When physical volumes are assigned to volume groups, they are *partitioned* into sub-units known as *physical partitions*. This partitioning increases the efficiency of disk operations. By default, these physical partitions are 4MB in size.

After creation, a volume group is broken down into *logical* sub-units known as *logical volumes*. Up to 256 logical volumes may be defined in a volume group. Logical volumes consist of *logical partitions,* which correspond to specific physical partitions. This strategy allows physical partitions from different physical volumes to be allocated to a logical volume.

Each logical volume contains a UNIX *filesystem*. It is to a filesystem that files are saved and retrieved. A certain number of physical partitions may be left unallocated after physical volume definition. This allows the system administrator to allocate those partitions to filesystems as needed, instead of having to "guesstimate" the needs of filesystems long before they are full. On initial configuration, partitions may be reserved for later allocation to filesystems as needed. Once added, allocated partitions may not be removed under normal administrative procedures. Thus, it is reasonable to reserve partitions to accommodate future growth.

Over time, available space in a filesystem decreases, as data is stored and files grow. However, the *logical* disk structure provided by UNIX allows the following benefits:

- If a system contains multiple physical volumes, unallocated partitions on any physical volume may be allocated to a given filesystem. The system will store and retrieve data to/from those partitions as if they were physically joined to that filesystem.

- If all physical volumes in a system are full, a new disk may be added. The physical partitions on the new disk may be allocated to any existing filesystem. Physical disk space is the total size of the disk drive(s) attached locally to a given system. Begin by determining the organization of your system disk. This permits the amount of available space to be determined and permits development of a plan to allocate that space over time. There are four steps to this process. You must log in as root (superuser) to use some of the commands found in these procedures. Identify the physical volumes (disks) in your system, then evaluate how physical disk space is used in your system:

    - Determine the logical definition of the physical volumes
    - Determine how much disk space is allocated to the filesystems
    - Calculate the available disk space

AIX Disk Analysis Procedure

Step One: Identifying Physical Volumes

1. Enter the *list physical volumes* command at the following prompt:

    # lspv

2. The system displays information similar to:

hdisk0 00297fb5608e4cf5 rootvg
hdisk1 001eb78425b9960f external9

3. The first column lists the names of the physical disks on the system, along with their *physical volume identifiers* (2nd column) and the *volume group names* to which each physical disk is assigned (3rd column). The first disk in the boot-up sequence, hdisk0, is assigned to the default root volume group rootvg; hdisk1 , an external 9GB disk drive, is assigned to the user-defined volume group external9.

4. Once the names of the hard drives are known, the overall size of the disk drive(s) may be evaluated. At the prompt, enter:

#lsdev -Cc disk

5. The system displays:

hdisk0 Available 04-B0-00-0,0 2.0 GB SCSI Disk Drive
hdisk1 Available 04-B0-00-1,0 Other SCSI Disk Drive

Step Two: Evaluate how Physical Disk Space is Used

1. Choose the disk you wish to evaluate. For example, if hdisk0 is to be evaluated, enter:

lspv hdisk0

2. The system responds:

```
PHYSICAL VOLUME: hdisk0              VOLUME GROUP: rootvg
PV IDENTIFIER: 00297fb5608e4cf5      VG IDENTIFIER:00297fb559b0ccc4
PV STATE: active
STALE PARTITIONS: 0                  ALLOCATABLE: yes
PP SIZE: 4 megabyte(s)               LOGICAL VOLUMES: 8
TOTAL PPs: 511 (2044 megabytes)      VG DESCRIPTORS: 2
FREE PPs: 320 (1280 megabytes)
USED PPs: 191 (764 megabytes)
FREE DISTRIBUTION:102..38..00..78..102
USED DISTRIBUTION:01..64..102..24..00
```

- PP SIZE indicates the size of the physical partitions. Although 4MB is the system default, each physical partition may be sized between 1 and 256 MB. Each physical volume may be divided into up to 1016 *physical partitions* (PPs).

- TOTAL PP's (above, 511) indicates how many physical partitions hdisk0 is divided into, and how much physical disk is included in these partitions.

- FREE PP's (above, 320) is the number of physical partitions currently unallocated to filesystems, and how much physical disk is included in these partitions.

- USED PP's (above, 191) is the number of physical partitions assigned to filesystems, and how much physical disk is included in these partitions.

- LOGICAL VOLUMES indicates how many filesystems are defined on hdisk0.

Step Three: Determine the Logical Definition of the Physical Volumes

1. The above example determined that eight logical volumes were defined on the volume group rootvg. These logical volumes may be displayed as follows:

lsvg %l rootvg

2. The system displays:

LVNAME	TYPE	LPs	PPs	PVs	LVSTATE	MOUNTPOINT
hd6	paging	64	64	1	open/syncd	N/A
hd5	boot	1	1	1	closed/sync d	N/A
hd8	jfslog	1	1	1	open/syncd	N/A
hd4	jfs	4	4	1	open/syncd	/
hd2	jfs	116	116	1	open/syncd	/usr
hd9var	jfs	1	1	1	open/syncd	/var
hd3	jfs	3	3	1	open/syncd	/tmp
hd1	jfs	1	1	1	open/syncd	/home

According to the output, rootvg has logical volumes hd1 through hd6, hd8 and hd9var defined. In a given system, most logical volumes will be associated with filesystems (*mount points*) accessible to the user, such as / (root), /var, /tmp, /usr, and /home. Mount points define how the logical volume's filesystem will be accessed. Accessible filesystems have the type jfs, which stands for *journaled file system* . Logical volumes with filesystems reserved for system use have other types; such volumes have mount points described as N/A.

Step Four: Determine Disk Space Allocated to Filesystems

1. Enter the command:

#df -k

2. The system displays the journaled file systems in the following format. Since only the journaled filesystems are displayed, the other filesystems listed in Step 3 using the lspv command must be taken into account when doing disk space calculations.

Filesystem	**1024-blocks**	**Free**	**%Used**	**Iused**	**%Iused**	**Mounted on**
/dev/hd4	16384	7548	54%	1035	13%	/
/dev/hd2	475136	37116	93%	20120	17%	/usr

/dev/hd9var	4096	1804	56%	150	15%	/var
/dev/hd3	12288	11472	7%	76	2%	/tmp
/dev/hd1	4096	3920	5%	19	2%	/home
/dev/lv00	3014656	269392	92%	15885	3%	/e2000
/dev/lv01	5308416	1633356	70%	25712	2%	/amc

- 1024 BLOCKS describes the number of 1K blocks allocated to the filesystem. The lv00 filesystem has, for example, 3,014,656 1K blocks allocated to it.
- FREE refers to the number of allocated 1K blocks not being used for data storage at this time.
- USED refers to the percentage of blocks allocated to the filesystem currently used for data storage. High percentages for highly active filesystems are cause for concern. The df command only gives part of the necessary information. We also know from the lsvg output that rootvg has an allocation of 66 physical partitions associated with hd5, hd6, and hd8. Since the physical partition size is 4MB (lspv output), 264MB are allocated to these special partitions.

Adding Unallocated Disk to a Filesystem Procedure

This procedure may be used to add unallocated space on a hard drive to a filesystem. When systems are initially set up, space may be reserved so that it may be added to the appropriate filesystem for later allocation. Once added to a filesystem, it may not be removed.

1. Determine PP SIZE and FREE PPs

% su
[root user password]
% lspv (displays the physical volume names on the system)
% lspv hdiskx x=disk# (displays how much disk space is utilized)

Example lspv output:

```
hdisk0          00297fb5608e4cf5    rootvg
hdisk1          001eb78425b9960f    external9
```

Example lspv hdiskx output. The PP SIZE value is the amount of disk available on the drive.

lspv hdisk0

```
PHYSICAL VOLUME:    hdisk0                  VOLUME GROUP:     rootvg
PV IDENTIFIER:      00297fb5608e4cf5        VG IDENTIFIER
00297fb559b0ccc4
PV STATE:           active
STALE PARTITIONS:   0                       ALLOCATABLE:      yes
PP SIZE:            4 megabyte(s)           LOGICAL VOLUMES:  8
TOTAL PPs:          511 (2044 megabytes)    VG DESCRIPTORS:   2
```



```
 FREE PPs:           320 (1280 megabytes)
USED PPs:           191 (764 megabytes)
FREE DISTRIBUTION:  102..38..00..78..102
USED DISTRIBUTION:  01..64..102..24..00
```

lspv hdisk1

```
PHYSICAL VOLUME:    hdisk1                  VOLUME GROUP:     external9
PV IDENTIFIER:      001eb78425b9960f        VG IDENTIFIER
00297fb5594d63a0
PV STATE:           active
STALE PARTITIONS:   0                       ALLOCATABLE:      yes
PP SIZE:            16 megabyte(s)          LOGICAL VOLUMES:  3
TOTAL PPs:          542 (8672 megabytes)    VG DESCRIPTORS:   2
 FREE PPs:           33 (528 megabytes)
USED PPs:           509 (8144 megabytes)
FREE DISTRIBUTION:  30..00..00..00..03
USED DISTRIBUTION:  79..108..108..108..106
```

2. Determine how much space is currently used in each of the filesystems. Use either:

```
% df -k [displays used space in 1024-byte blocks]
% df    [displays used space in 512-byte blocks
```

Example df -k output:

```
Filesystem    1024-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4            16384      7548   54%     1035    13% /
/dev/hd2           475136     37116   93%    20120    17% /usr
/dev/hd9var          4096      1804   56%      150    15% /var
/dev/hd3            12288     11472    7%       76     2% /tmp
/dev/hd1             4096      3920    5%       19     2% /home
/dev/lv00         3014656    269392   92%    15885     3% /e2000
/dev/lv01         5308416   1633356   70%    25712     2% /amc
```

Example df output:

```
 Filesystem    512-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4            32768     15096   54%     1035    13% /
/dev/hd2           950272     74232   93%    20120    17% /usr
/dev/hd9var          8192      3608   56%      150    15% /var
/dev/hd3            24576     22944    7%       76     2% /tmp
/dev/hd1             8192      7840    5%       19     2% /home
/dev/lv00         6029312    538784   92%    15885     3% /e2000
/dev/lv01        10616832   3266712   70%    25712     2% /amc
```

3. Decide on the new allocation

Based on the information given by the df command, determine the number of blocks you wish to add to each of the filesystems to be increased. If an error meesage related to the size of logical volumes appears, proceed to step 4, then return to step 3.

4. Enlarge the filesystem

To increase a filesystem, enter:

chfs -a size=[desired size in 1024K blocks] [volume the filesystem to be enlarged is mounted on]

5. Review the change by using the df command.

Occasionally, a error will result due to the default partitioning of the logical drives. If such an error occurs, preventing the filesystem from being sized, change the logical volume partitioning by:

chlv -x [number of partitions] [filesystem to be sized]

-

