– LPI 101 –

Maintain the integrity of filesystems

(Linux Professional Institute Certification)

[a]

```
    .~.
    /V\     by: geoffrey robertson
   // \\         geoffrey@zip.com.au
   @._.@
```

```
$Id: gl1.104.2.slides.tex,v 1.3 2003/05/30 05:04:30 waratah Exp $
```

Maintain the integrity of filesystems

Objective

Candidates should be able to verify the integrity of filesystems, monitor free space and inodes, and repair simple filesystem problems. This objective includes the commands required to maintain a standard filesystem, as well as the extra data associated with a *journaling* filesystem.

Key files, terms, and utilities include:

```
du    df    fsck   e2fsck    mke2fs
debugfs    dumpe2fs    tune2fs
```

Summary of Commands

Command	*Function*
du	Display disk usage
df	Display disk space free
fsck	Check Filesystem
e2fsck	Check an ext2 Filesystem
mke2fs	Create an ext2 Filesystem
debugfs	Debug an ext2 Filesystem
dumpe2fs	Dump Filesystem information
tune2fs	Adjust parameters on ext2 Filesystem

du - Disk Usage

- du shows *disk usage*
- du can work with subdirectories or an entire disk
- Usage is:

  ```
  du [options] [directory]
  ```

Options to du

Option	*Function*
-a	Show counts for all files & directories
-b	Display size in bytes
-c	Print total for all arguments after processing
-h	Print in *human readable* form
-k	Show size in Kilobytes
-m	Display size in Megabytes
-s	Display a summary for each argument
-x	Skip directories containing other filesystems

df - Disk Free

- df shows disk space used & available
- Default is to display all filesystems
- Usage is:

  ```
  df [options] [directory]
  ```

Options to df

Option	*Function*
-a	Show counts for all filesystems
-t *fs type*	Limit listing to *fs type*
-h	Print in *human readable* form
-k	Show size in Kilobytes
-m	Display size in Megabytes
-i	Display inode information
-l	Limit listing to local filesystems
-x *fs type*	Exclude *fs type* from listing

fsck - Check and repair a Linux file system

- fsck is used to check and optionally repair a one or more Linux file systems.
- filesys can be a device name (e.g. /dev/hdc1, /dev/sdb2), a mount point (e.g. /, /usr, /home), or an ext2 label.
- fsck will try to run filesystems on different physical drives in parallel to reduce total amount time to check all of the filesystems.
- fsck makes 5 passes on the filesystem:
  - Pass 1: Check inodes, blocks & sizes
  - Pass 2: Check directory structure
  - Pass 3: Check directory connectivity
  - Pass 4: Check reference counts

- Pass 5: Check group summary information

fsck Options

Option	*Function*
-p	Automatically repair without prompting
-n	Don’t make changes to filesystem
-y	Assume yes to all questions
-f	Force check even if fs is clean
-r	Interactively prompt for changes
-v	Be verbose
-A	Check all filesystems in `/etc/fstab`
-C	Display a progress bar
-N	Don’t execute, show what would be done

fsck error codes

When fsck completes, it will return a value ($?) as follows:

Code	*Meaning*
0	No errors
1	Errors found & corrected
2	System should be rebooted
4	Filesystem error left uncorrected
8	Operational error
16	Usage or syntax error
128	Shared library error

e2fsck - Check a Linux ext2 FS

- e2fsck is used to check a Linux second extended file system (e2fs)
- e2fsck also supports ext2 filesystems containing a journal, which are also sometimes known as ext3 filesystems.
- e2fsck operates in a similar manner to fsck (see man page)

mke2fs - Create a Linux ext2 filesystem

- mke2fs is used to create an ext2 filesystem
- mke2fs takes a device special file as its argument
- mkfs.ext2 is the same as mke2fs
- To make an ext3 filesystem, you first make an ext2 filesystem and then add a journal to it using tune2fs or use the `-j` option to mke2fs
- Usage is:

```
mke2fs [options] device
```

mke2fs Options

Option	*Use*
-V	Be verbose
-b *blocksize*	Make blocks *blocksize* bytes
-c	Check for bad blocks on device
-i *bytes per inode*	Create an inode for each *bytes per inode*
-j	Create a journal (ext3)
-L *label*	Set the volume label
-N *inodes*	Create the fs with specified number of inodes
-n	Show what would be done (don’t actually create fs)

debugfs - Ext2 filesystem debugger

- debugfs is a file system debugger
- It can be used to examine and change the state of an ext2 file system.
- debugfs is an interactive debugger. It understands a number of commands:

  cat filespec Dump the contents of the inode filespec to stdout.

  cd filespec Change the current working directory to filespec.

  chroot filespec Change the root directory to be the directory file spec.

  close Close the currently open file system.

  quit Exit debugfs.

dumpe2fs - Dump filesystem information

- dumpe2fs prints the super block and blocks group information for the filesystem present on device
- Usage is:

```
dumpe2fs [options] device
```

- Common options are:

Option	*Use*
-b	Display badblocks on device
-h	Display superblock information

tune2fs - Adjust filesystem parameters on ext2 fs

- tune2fs adjusts tunable filesystem parameters on a Linux ext2 filesystem.
- tune2fs can be used to add a journal to an ext2 filesystem.
- Usage is:

  ```
  tune2fs [options] device
  ```

tune2fs - Common options

Option	*Use*
-c *max-mounts*	Set no of mounts before fsck is forced
-g *group*	Set the group who can use reserved blocks
-j	Add a journal to the filesystem
-L *label*	Set the volume label
-r *blocks*	Set the number of reserved blocks