File Management Tools

- gzip and gunzip
- tar
- find
- df and du
- od
- sftp and scp

Gzip and Gunzip

- The *gzip* utility compresses a specified list of files. After compressing each specified file, it renames it to have a ".gz" extension.
- General form.

  gzip [*filename*]*
- The *gunzip* utility uncompresses a specified list of files that had been previously compressed with gzip.
- General form.

  gunzip [*filename*]*

Tar (38.2)

- *Tar* is a utility for creating and extracting archives. It was originally setup for archives on tape, but it now is mostly used for archives on disk. It is very useful for sending a set of files to someone over the network. *Tar* is also useful for making backups.
- General form.

  tar *options filenames*

Commonly Used Tar Options

```
c    # insert files into a tar file
f    # use the name of the tar file that is specified
v    # output the name of each file as it is inserted into or
     # extracted from a tar file
x    # extract the files from a tar file
```

Creating an Archive with Tar

- Below is the typical *tar* command used to create an archive from a set of files. Note that each specified filename can also be a directory. *Tar* will insert all files in that directory and any subdirectories.

  tar cvf *tarfilename filenames*

- Examples:

```
tar cvf proj.tar proj      # insert proj directory
                           # files into proj.tar
tar cvf code.tar *.c *.h   # insert *.c and *.h files
                           # into code.tar
```

Extracting Files from a Tar Archive

- Below is the typical *tar* command used to extract the files from a *tar* archive. The extracted files will have the same name, permissions, and directory structure.

  tar xvf *tarfilename*

- Examples:

```
tar xvf proj.tar   # extract files from proj.tar
tar xvf code.tar   # extract files from code.tar
```

Find (9.1)

- The *find* unix utility recursively searches within a directory for files that match specified attributes.
- General form. *Find* recursively searches in the specified directories for the files that match the specified operators.

  find [*directory*]+ *operators*

Find Operators (9.1)

-name *filename* # find files with the specified filename,
if wildcard characters, then uses quotes
-size *n*[c] # find files that are over *n* blocks in size,
where a block is 512 bytes, the c
# indicates bytes
-atime *n* # find files accessed over *n* days ago
-mtime *n* # find files modified over *n* days ago
-print # prints name of the file (typical default)
-exec *command* # execute *command* for each file that
matches, *command* must end with "\;",
can refer to file that matched using "{}"
-okay *command* # same as -exec, but user is prompted for
for permission to perform the command

Find Operators (cont.)

oper1 -a *oper2* # find files that match *oper1* **and** *oper2*
oper1 -o *oper2* # find files that match *oper1* **or** *oper2*
!*oper* # find files that do **not** match *oper*
\(*expr* \) # evaluate *expr* before other operators
that are outside of the parentheses

Find Examples

```
find .                # print all files in current directory
                      # and all subdirectories
find /tmp -mtime 7    # find all files in /tmp that have not
                      # been modified in over 7 days
find . -name core -exec rm {} \;
                      # remove all core files found
find . -name core -o -name “*.o” -okay rm {} \;
                      # remove all core and *.o files found,
                      # but check confirmation from the user
find . -name “*.sh” -exec chmod +x {} \;
                      # add execute permission on all *.sh
                      # files
grep main `find . -name *.c`
                      # find all *.c files including
                      # subdirectories and grep for the main
                      # function in them
```

Df and Du

- The *df* command displays the amount of disk used and available on all mounted file systems. If you specify a filename, then it shows disk space that is available on the file system containing the file.

  df [*filenames*]

- The *du* command writes the size of each directory and subdirectory in 1024 byte units. By default it uses the current directory. However, you can specify the set of files and directories for it to inspect. The -s option indicates to just print the total sum and not information for each file.

  du [-s] [filenames]

Od (12.4)

- The *od* command copies each input file to standard output and transforms the input data according to the output types specified in the options given in the command. This utility is often used for finding special embedded control characters in a file.
- General form.

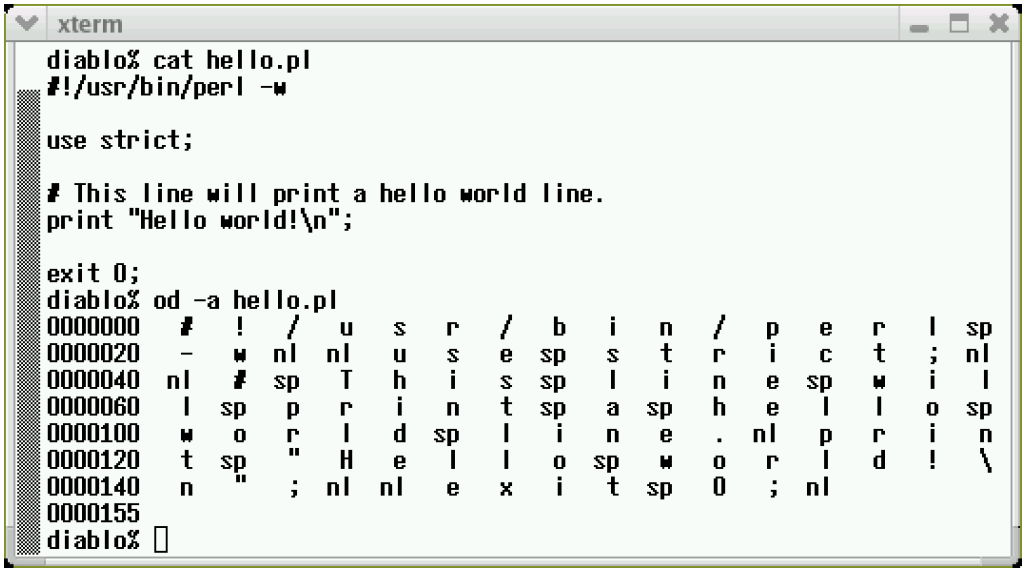
  od [*options*] [*filenames*]

Od Options

- By default, *od* prints words out in octal. In fact, *od* stands for octal dump. Below are some commonly used od options.

```
-a    # prints bytes as characters, special values are
      # listed using character codes (newline => nl)
-b    # prints bytes as octal values
-c    # prints bytes as characters, some special values
      # are listed as C escapes (newline => \n), others
      # are listed as octal values
```

Od Example



Nm: Print Name List of Object Files

- The *nm* utility prints the global names of functions and variables within object files or executables.
- These names are used by a linker and the loader to resolve external references.
- This feature is useful to determine where specific routines or variables are defined when you do not have access to the source code.

nm [*options*] *files*

Strip: Remove Optional Information from Object Files

- The *strip* unix utility removes optional symbol table, debugging, and line number information from an object file or executable. *Strip* is used to reduce the file storage required for these files. It can also be used to make an executable more secure.

strip *files*

Sftp: Secure File Transfer Program

- The *sftp* utility can be used to transfer files over the network between machines using encrypted communication. After issuing the *sft*p command, sftp will prompt you for a password on the remote machine.
- General form.

  sftp [*hostname* | *user@hostname*]

Commonly Used Sftp Commands

get *remotefile* [*localfile*]	# transfer file from remote site
put *localfile* [*remotefile*]	# transfer file to remote site
cd *path*	# change remote directory to *path*
lcd *path*	# change local directory to *path*
chmod *mode file*	# change permissions on *file* to *mode*
pwd	# display remote working directory
lpwd	# display local working directory
mkdir *path*	# create remote directory

Scp: Secure Copy

- The *scp* utility allows you to copy files to/from a remote machine using encrypted communication. After issuing the command, *scp* will prompt you for a password on the remote machine.

  # transfering a file to a remote machine
  scp *localfile user@hostname*:*remotefile*

  # transfering a file from a remote machine
  scp *user@hostname*:*remotefile localfile*