

TITAN USERS GUIDE

Mike Dixon
Research Applications Laboratory
National Center for Atmospheric Research
Boulder Colorado USA

October 2005

UNIX basics

Before launching into the details of how to use the TITAN System, it is useful to develop some familiarity with the foundations upon which TITAN is built. UNIX is an operating system distinct from other operating systems which come from Microsoft.

UNIX vs Windows

	Variations	Distributors
UNIX	Linux	Redhat, Mandrake, Debian, etc.
	IRIX	SGI
	Solaris	SUN
	AIX	IBM
Microsoft	DOS	Microsoft
	Windows	Microsoft
	NT	Microsoft

Although UNIX and Microsoft operating systems are distinct, there are similarities between them. One similarity is that in both UNIX and in Microsoft DOS one can execute commands in a window. Another similarity is that both have a hierarchical file system structure with partitions, directories, sub-directories, and files. Thus, the command for moving to a directory location is similar in both operating systems:

	Command	Example
UNIX	cd	cd /home/titan5
Microsoft DOS	cd	cd C:\titan5

To view the contents of a directory:

	Command	Example
UNIX	ls	ls /home/titan5
Microsoft DOS	dir	dir C:\titan5

NOTE: UNIX uses a forward slash while DOS uses a backward slash in directory paths.

Starting UNIX

For various reasons, the TITAN computers should continue running 24 hours a day. It is **important** that you do not turn off a UNIX computer without first following the appropriate shutdown procedures. If, for some reason, the TITAN computer is off (like at the beginning of this training class), simply power on the computer to startup UNIX.

To Begin a User Session

After turning on the computer a login prompt will appear. Like Microsoft Windows you must enter a user name and password. The user name for the TITAN System is **titan5**. In the training class you will learn the password for the titan5 user. For the security of the DPC radar network, the password is not written here; but it is important that each TITAN user know the titan5 password.

To Enter UNIX Commands

After logging into UNIX you will see a window for entering operating system commands. This type of window is called an **xterm**. Unlike Microsoft DOS, UNIX is case sensitive. Thus the command `ls` cannot be typed as `LS`.

When you first login to the UNIX operating system, you are put into a directory that is called your **home** directory. Unlike Microsoft Windows, each user has their own unique home directory. For example, on the computers used in this training class the home directory for the titan5 user is **/home/titan5**.

In the home directory for the titan5 user, there are various files and sub-directories that you can see using the `ls` command. One sub-directory in **/home/titan5** is **san_rafael**. You can view the contents of this sub-directory in various ways:

The command `ls san_rafael` reveals the contents of the sub-directory. In this usage **/home/titan5** is the current directory and **san_rafael** is considered a **relative path** because it is a sub-directory that is relative to the current directory location.

The command `ls /home/titan5/san_rafael` will also reveal the contents of the same sub-directory. This command form makes use of an **explicit path**, i.e., the **san_rafael** sub-directory is fully qualified and is not relative to the current directory location. Note that the explicit path begins with a forward slash while the relative path (in example 1 above) does not.

Finally, you can view the contents of the same sub-directory by using two commands: `cd san_rafael` followed by `ls`.

Command line arguments

In UNIX, commands typed on the command line are most often qualified by one or more command line arguments. These arguments start with a '-' character.

For example, the command:

```
top
```

will start the top application which lists a table of running processes, along with information on other aspects of the state of the UNIX operating system. Top will continue running and updating its table until you kill it with a control-c keystroke.

Top may be run in a slightly different manner, however. If you run:

```
top -n 1
```

top will write its table once and then exit. The '-n' is referred to as a command line argument, and the '1' is the value of that argument.

Command line arguments may be strung together in a list. For example:

```
rm -r -f -v /tmp/junk
```

removes the directory /tmp/junk. The -f argument tells rm to not prompt the user if the directory does not exist and the -v argument tells rm to print details of what it is doing.

You can discover the available list of arguments for commands in a number of ways. The 'man' or 'manual pages' for a command will list the available arguments - see the next section. You can also often prompt the application to list its command line arguments by giving it the '-h' or '--' command line argument. All TITAN applications respond to the -h command line argument.

Man (manual) pages

For example, the following is a simple man page for the rm command:

RM(1)
Commands

User
RM(1)

NAME

rm - remove files or directories

SYNOPSIS

rm [OPTION]... FILE...

DESCRIPTION

This manual page documents the GNU version of rm. rm removes each specified file. By default, it does not remove directories.

If a file is unwritable, the standard input is a tty, and the -f or --force option is not given, rm prompts the user for whether to remove the file.

If the response does not begin with 'y' or 'Y', the file is skipped.

OPTIONS

Remove (unlink) the FILE(s).

-d, --directory

unlink FILE, even if it is a non-empty directory (super-user only; this works only if your system

supports 'unlink' for nonempty directories)

-f, --force

ignore nonexistent files, never prompt

-i, --interactive

prompt before any removal

--no-preserve-root do not treat '/' specially (the default)

--preserve-root

fail to operate recursively on '/'

-r, -R, --recursive

remove the contents of directories recursively

-v, --verbose

explain what is being done

--help display this help and exit

--version

output version information and exit

To remove a file whose name starts with a '-', for example '-foo', use one of these commands:

rm -- -foo

rm ./-foo

Note that if you use rm to remove a file, it is usually possible to recover the contents of that file. If you want more assurance that the contents

are truly unrecoverable, consider using shred.

AUTHOR

Written by Paul Rubin, David MacKenzie, Richard Stallman, and Jim Meyering.

REPORTING BUGS

Report bugs to <bug-coreutils@gnu.org>.

COPYRIGHT

Copyright (C) 2004 Free Software Foundation, Inc.

```
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.

SEE ALSO
    chattr(1), shred(1)
```

The full documentation for rm is maintained as a Texinfo manual. If the info and rm programs are properly installed at your site, the command

The File Structure

Other commands convenient for moving around the UNIX file system are:

```
cd      to return to the user's home directory
cd ..   to go up one directory level
pwd     to find out the location of the current directory
```

Furthermore, it is possible to see more detail about the contents of a directory by using various options on the `ls` command. For example:

```
ls -l to show information such as the date, size, and owner of each file
ls -F to show information about the type of each file
```

In addition to information such as file size, the command `ls -l` reveals a special notation called a **symbolic link**. For example in the home directory for titan5 there are two symbolic links: `titan_home` and `data`. Symbolic links in UNIX, like shortcuts in Microsoft Windows, are simply quick routes to a specific directory or file.

The command `ls -F` does not show a lot of detail about each file, but the `-F` option is used to display simple notation about the types of files in a directory. For example, when using the `ls -F` command:

```
file1/ indicates that file1 is a sub-directory
file2@ indicates that file2 is a symbolic link
file3* indicates that file3 is executable
```

NOTE: In UNIX the names of files and directories can have more than 8 characters. Also file extensions do not change the behavior of a file and are only included for information purposes. For example, executables do not need to have `.exe` in their file names.

Making Changes

Some useful commands for changing files in a directory are:

```
cp <file1> <file2> to copy file1 to file2
```

```
mv <file1> <file2> to move file1 to file2
rm <file1> to erase file1
```

Furthermore it is important that you learn to use a text editor, for example **xedit** or **kedit**, for making changes to the contents of the files themselves.

Environment Variables

In UNIX it is possible to use variables to hold values which are used later when executing commands. These types of variables are called **environment variables**. The **\$** is required when using an environment variable in a UNIX command, and by convention environment variables are written in uppercase. For example in the TITAN System is defined an environment variable **\$TITAN_HOME** which is set to the directory where the TITAN configuration files reside. On the computers in the training class, **\$TITAN_HOME** expands to **/home/titan5/titan_home**.

In this case since **\$TITAN_HOME** is a directory location, the following commands have the same results:

```
cd $TITAN_HOME
cd /home/titan5/titan_home
```

To see the current value of an environment variable, use the **echo** command. For example:

```
echo $TITAN_HOME
```

Another important environment variable used in the TITAN System is **\$DATA_DIR** which expands to a directory location where all of the data for the TITAN System are stored.

To see the full list of defined environment variables and their values, use the command:

```
printenv
```

From the **printenv** command you will see many environment variables which have been defined for the titan5 user. Some of these environment variables are defined by the TITAN System, while others are defined by the UNIX operating system.

One environment variable which UNIX defines for each user is **\$HOME**. Depending on how the UNIX file partitions were configured when UNIX was installed, the location of the user's home directory may differ. For example, on the computers in this training class, **\$HOME** for the titan5 user expands to **/home/titan5**. However, on the DPC TITAN System computers, **\$HOME** expands to **/hd/titan5**.