File handling in TCL

File handling in Tcl (Tool Command Language) involves a set of commands that allow you to create, read, write, and manipulate files and directories. Below is a detailed overview of the key operations you can perform for file handling in Tcl:

1. Opening Files

To work with a file, you first need to open it using the open command. You can specify the mode in which you want to open the file:

- **Read Mode (r)**: Open a file for reading.
- **Write Mode (w)**: Open a file for writing (creates a new file or truncates an existing one).
- **Append Mode (a)**: Open a file for appending (writes data at the end of the file).
- **(Read/Write Mode) (r+)**

  Description: Opens a file for both reading and writing. The file must already exist; if it does not, an error will occur.

  Behavior: The file pointer is positioned at the beginning of the file. You can read from and write to the file, but writing will overwrite existing content starting from the current file pointer position.

(Write/Read Mode) (w+)

Description: Opens a file for both writing and reading. If the file already exists, it is truncated to zero length (i.e., all existing content is deleted). If the file does not exist, a new file is created.

Behavior: The file pointer is positioned at the beginning of the file. You can write to the file, and any existing content will be lost. After writing, you can read from the file, but you may need to reposition the file pointer using seek.

(Append/Read Mode) (a+)

Description: Opens a file for both appending and reading. If the file does not exist, a new file is created. If it does exist, the file pointer is positioned at the end of the file for writing.

Behavior: You can read from the file, and when you write, the new content is added at the end of the file without overwriting existing content. If you want to read from the beginning, you need to reposition the file pointer.

- **`r,r+`**: Read and (Read /write); file must exist; does not truncate.
- **`w,w+`**: write and (Write and read); truncates existing file or creates a new one.
- **`a,a+`**: Append and (Append and read); creates a new file if it doesn't exist; appends to the end.

1. **Binary Mode**:
   - You can append "b" to any of the above modes to open the file in binary mode.
   - This is useful for reading or writing binary data (e.g., images, executables).

   For example:

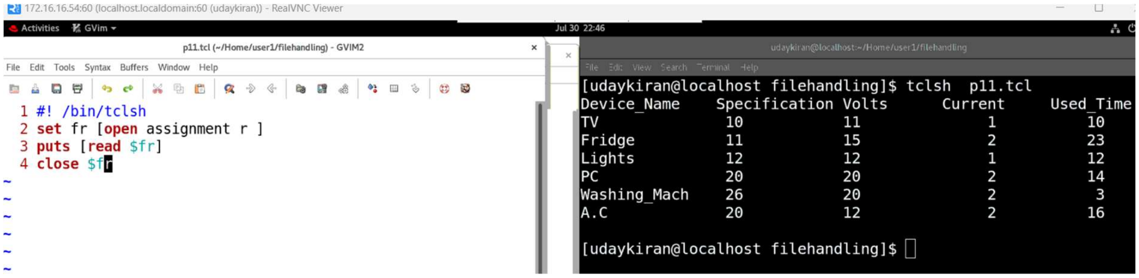
Open for reading in binary mode:

```
set fileId [open "example.bin" "rb"]
```

Open for writing in binary mode:

```
set fileId [open "example.bin" "wb"]
```

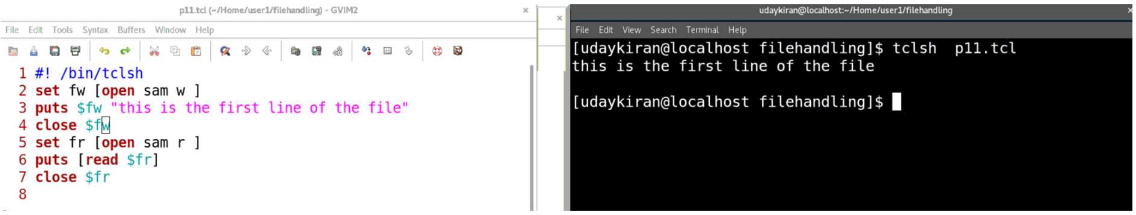
reading a file content:


172.16.16.54:60 (localhost.localdomain:60 (udaykiran)) - RealVNC Viewer
Activities
GVim
Jul 30 22:46
p11.tcl (~/Home/user1/filehandling) - GVIM2
File Edit Tools Syntax Buffers Window Help
udaykiran@localhost:~/Home/user1/filehandling
File Edit View Search Terminal Help

```
#! /bin/tclsh
set fr [open assignment r ]
puts [read $fr]
close $fr
```

```
[udaykiran@localhost filehandling]$ tclsh  p11.tcl
Device_Name    Specification Volts      Current      Used_Time
TV              10            11           1          10
Fridge          11            15           2          23
Lights          12            12           1          12
PC              20            20           2          14
Washing_Mach    26            20           2           3
A.C             20            12           2          16

[udaykiran@localhost filehandling]$
```

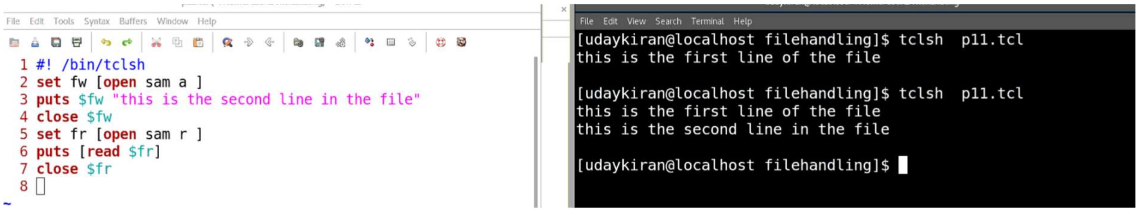
Writing content in file:


p11.tcl (~/Home/user1/filehandling) - GVIM2
File Edit Tools Syntax Buffers Window Help
udaykiran@localhost:~/Home/user1/filehandling
File Edit View Search Terminal Help

```
#! /bin/tclsh
set fw [open sam w ]
puts $fw "this is the first line of the file"
close $fw
set fr [open sam r ]
puts [read $fr]
close $fr

```

```
[udaykiran@localhost filehandling]$ tclsh  p11.tcl
this is the first line of the file

[udaykiran@localhost filehandling]$
```

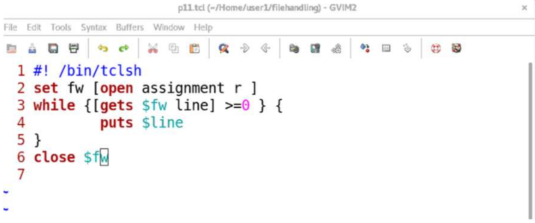
Appending content in the file:


File Edit Tools Syntax Buffers Window Help
File Edit View Search Terminal Help

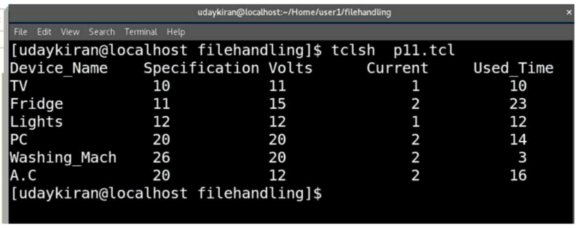
```
#! /bin/tclsh
set fw [open sam a ]
puts $fw "this is the second line in the file"
close $fw
set fr [open sam r ]
puts [read $fr]
close $fr

```

```
[udaykiran@localhost filehandling]$ tclsh  p11.tcl
this is the first line of the file

[udaykiran@localhost filehandling]$ tclsh  p11.tcl
this is the first line of the file
this is the second line in the file

[udaykiran@localhost filehandling]$
```

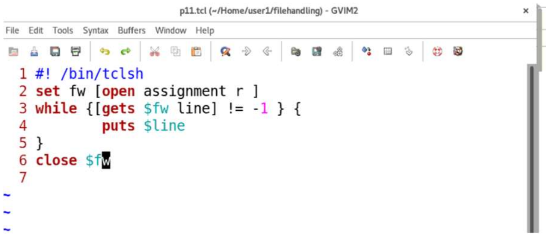
Reading the content in the file line by line:


p11.tcl (~/Home/user1/filehandling) - GVIM2
File Edit Tools Syntax Buffers Window Help

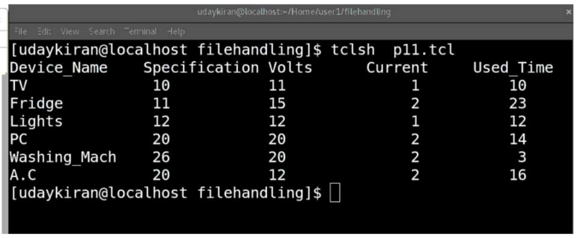
```
#! /bin/tclsh
set fw [open assignment r ]
while {[gets $fw line] >=0 } {
          puts $line
}
close $fw
```


udaykiran@localhost:~/Home/user1/filehandling
File Edit View Search Terminal Help

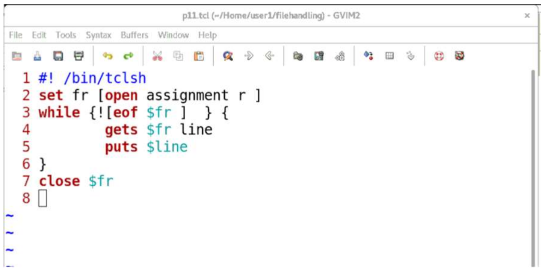
```
[udaykiran@localhost filehandling]$ tclsh  p11.tcl
Device_Name     Specification Volts       Current     Used_Time
TV               10            11            1           10
Fridge           11            15            2           23
Lights           12            12            1           12
PC               20            20            2           14
Washing_Mach     26            20            2            3
A.C              20            12            2           16
[udaykiran@localhost filehandling]$
```


p11.tcl (~/Home/user1/filehandling) - GVIM2
File Edit Tools Syntax Buffers Window Help

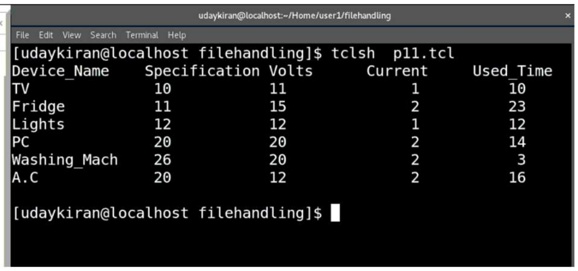
```
#! /bin/tclsh
set fw [open assignment r ]
while {[gets $fw line] != -1 } {
          puts $line
}
close $fw
```


udaykiran@localhost:~/Home/user1/filehandling
File Edit View Search Terminal Help

```
[udaykiran@localhost filehandling]$ tclsh  p11.tcl
Device_Name     Specification Volts       Current     Used_Time
TV               10            11            1           10
Fridge           11            15            2           23
Lights           12            12            1           12
PC               20            20            2           14
Washing_Mach     26            20            2            3
A.C              20            12            2           16
[udaykiran@localhost filehandling]$
```

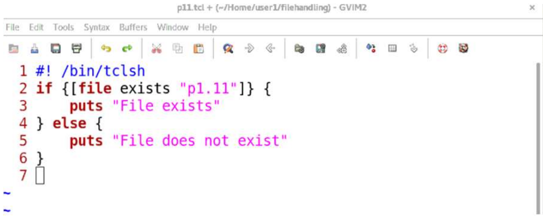

p11.tcl (~/Home/user1/filehandling) - GVIM2
File Edit Tools Syntax Buffers Window Help

```
#! /bin/tclsh
set fr [open assignment r ]
while {![eof $fr ]  } {
          gets $fr line
          puts $line
}
close $fr
```

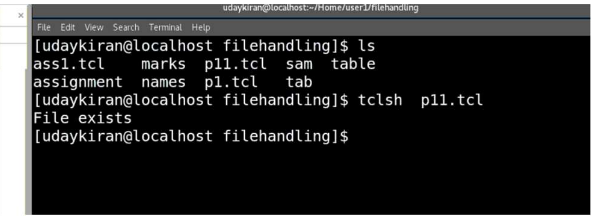

udaykiran@localhost:~/Home/user1/filehandling
File Edit View Search Terminal Help

```
[udaykiran@localhost filehandling]$ tclsh  p11.tcl
Device_Name     Specification Volts       Current     Used_Time
TV               10            11            1           10
Fridge           11            15            2           23
Lights           12            12            1           12
PC               20            20            2           14
Washing_Mach     26            20            2            3
A.C              20            12            2           16

[udaykiran@localhost filehandling]$
```

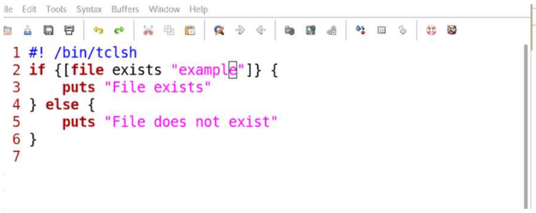
Checking whether file exits or not:


p11.tcl + (~/Home/user1/filehandling) - GVIM2
File Edit Tools Syntax Buffers Window Help

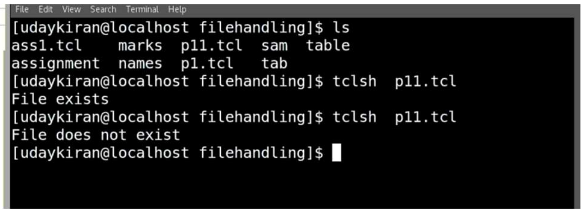
```
#! /bin/tclsh
if {[file exists "p1.11"]} {
     puts "File exists"
} else {
     puts "File does not exist"
}
```


udaykiran@localhost:~/Home/user1/filehandling
File Edit View Search Terminal Help

```
[udaykiran@localhost filehandling]$ ls
ass1.tcl    marks  p11.tcl  sam  table
assignment  names  p1.tcl   tab
[udaykiran@localhost filehandling]$ tclsh  p11.tcl
File exists
[udaykiran@localhost filehandling]$
```

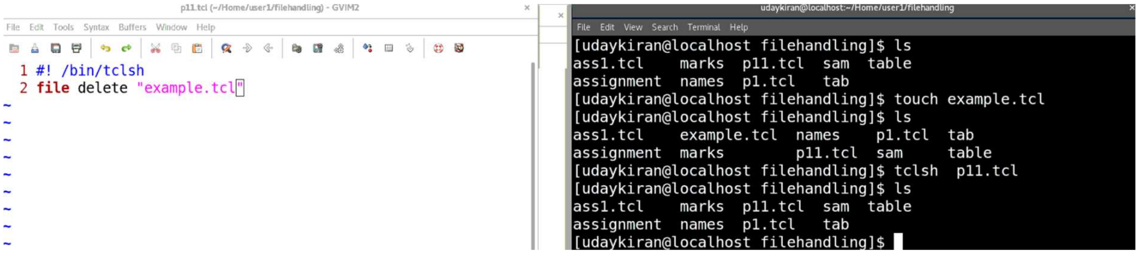


```
#! /bin/tclsh
if {[file exists "example"]} {
     puts "File exists"
} else {
     puts "File does not exist"
}
```



```
[udaykiran@localhost filehandling]$ ls
ass1.tcl    marks  p11.tcl  sam  table
assignment  names  p1.tcl   tab
[udaykiran@localhost filehandling]$ tclsh  p11.tcl
File exists
[udaykiran@localhost filehandling]$ tclsh  p11.tcl
File does not exist
[udaykiran@localhost filehandling]$
```

Deleting a file:


p11.tcl (~/Home/user1/filehandling) - GVIM2
File Edit Tools Syntax Buffers Window Help
udaykiran@localhost:~/Home/user1/filehandling
File Edit View Search Terminal Help

```
#! /bin/tclsh
file delete "example.tcl"
```

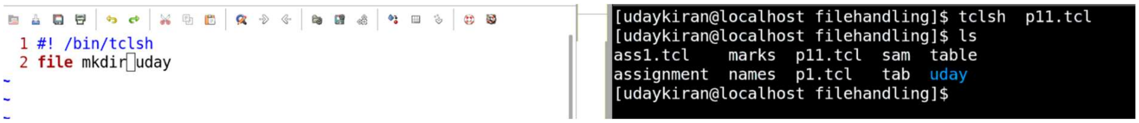
```
[udaykiran@localhost filehandling]$ ls
ass1.tcl    marks  p11.tcl  sam  table
assignment  names  p1.tcl   tab
[udaykiran@localhost filehandling]$ touch example.tcl
[udaykiran@localhost filehandling]$ ls
ass1.tcl    example.tcl  names    p1.tcl  tab
assignment  marks        p11.tcl  sam     table
[udaykiran@localhost filehandling]$ tclsh  p11.tcl
[udaykiran@localhost filehandling]$ ls
ass1.tcl    marks  p11.tcl  sam  table
assignment  names  p1.tcl   tab
[udaykiran@localhost filehandling]$
```

creating directories using the **file mkdir** command. This command is part of the `file` command suite, which provides various file and directory manipulation capabilities. Here's how you can create a directory and handle potential errors:

Basic Directory Creation

To create a directory, use the **file mkdir** command:

```
file mkdir new_directory
```



```
#! /bin/tclsh
file mkdir uday
```

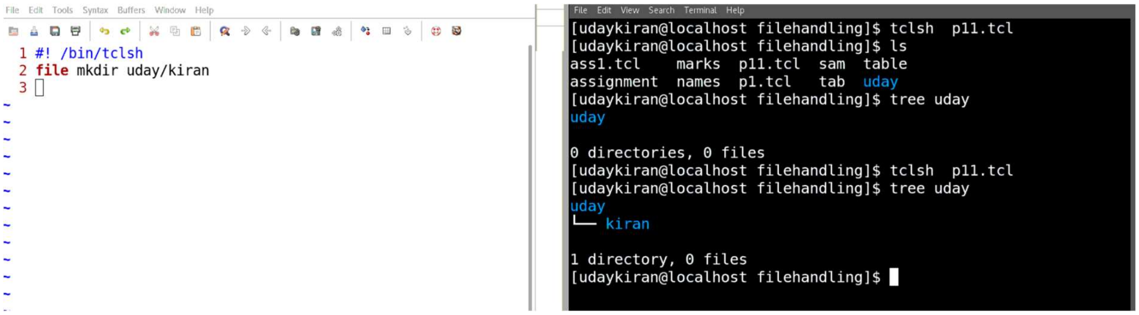
```
[udaykiran@localhost filehandling]$ tclsh  p11.tcl
[udaykiran@localhost filehandling]$ ls
ass1.tcl    marks  p11.tcl  sam  table
assignment  names  p1.tcl   tab  uday
[udaykiran@localhost filehandling]$
```

This command will create a directory named `new_directory` in the current working directory.

Creating Nested Directories

If you want to create nested directories, `file mkdir` will handle this for you as well:

```
file mkdir parent_directory/child_directory
```


File Edit Tools Syntax Buffers Window Help
File Edit View Search Terminal Help

```
#! /bin/tclsh
file mkdir uday/kiran

```

```
[udaykiran@localhost filehandling]$ tclsh  p11.tcl
[udaykiran@localhost filehandling]$ ls
ass1.tcl    marks  p11.tcl  sam  table
assignment  names  p1.tcl   tab  uday
[udaykiran@localhost filehandling]$ tree uday
uday

0 directories, 0 files
[udaykiran@localhost filehandling]$ tclsh  p11.tcl
[udaykiran@localhost filehandling]$ tree uday
uday
└── kiran

1 directory, 0 files
[udaykiran@localhost filehandling]$
```

This command creates `parent_directory` if it doesn't exist and then creates `child_directory` inside it.