# Lecture # 2.3
# Design and Code of
# UNIX more utility

## Course: Advanced Operating System
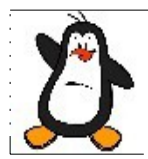
## Instructor: Arif Butt

## Punjab University College of Information Technology (PUCIT)
## University of the Punjab

# Agenda

- File Handling using Standard C Library functions
- Examples of Disk I/O using Library functions
- UNIX `more` utility
  - What does it do?
  - How does it do it?
  - Can we write our own?
  - Coding different versions of our own `more` utility

# I/O functions of Standard C Library

| Library Functions | Description |
|---|---|
| **fopen(), fclose()** | Opening and closing a file |
| **fgetc(), fputc()** | Reading/writing one character at a time |
| **fgets(), fputs()** | Reading/writing line by line |
| **fscanf(), fprintf()** | Reading/writing using formatted I/O |
| **fseek(), ftell(), fgetpos(), fsetpos() rewind()** | Getting information or changing the current file offset of a file |

# Examples: Disk IO

# UNIX `more` utility
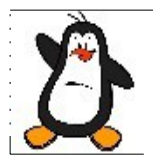
# How does more do it?

1. Show P-1 lines from file passed as cmd line arg
2. Show `--More--(54%)` message after P-1 lines
3. Wait for commands
4. If command is `'\n'` character, advance one line and got to step 2
5. If command is `' '` character, got to step 1
6. If command is `'q'` character, exit

# **morev0.c**

**Get a single file name via command line, display entire contents of file and exits**

# **`morev1.c`**

**Get multiple file names via command line, display contents of all the files and exits**

# morev2.c

**Display one page at a time and pause for commands (`'q'`,`' '`,`'\n'`) and act accordingly**
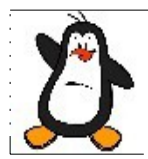
# A bit about Terminals

Terminal is a device used to communicate with a process. It consists of a keyboard and a Visual display unit. We have used `printf()` function to display text on standard output, and we know that the first argument to `printf` is a format string which can contain:

- Text to be displayed
- Escape sequence characters
- Format specifiers

We can also use Control Sequence Introducer (**\033[**), and after this we can use a specific Terminal Control Code, using which we can change how the text is displayed on the screen, can control the movement of the cursor as well as can clear parts of the screen

# Some important Control Codes

## To display text:

- **m** - Reset to default settings
- **1m** - Bold characters
- **2m** - Under line characters
- **7m** - Reverse video

## Foreground & Background colors:

- **30-39m** – Foreground colors
- **40-49m** - Background colors

**30m** - Default foreground color

**40m** - Default background color

## Cursor movements:

- **nA** - Move up **n** lines
- **nB** - Move down **n** lines
- **nC** - Move right **n** columns
- **nD** - Move left **n** columns
- **nG** - Move to column **n**
- **r,cH** - Move to row, column

## Clearing parts of screen:

- **0K**-Erase from cursor to end of line
- **1K**-Erase from start of line to cursor
- **2K**-Erase current line
- **0J**-Erase from cursor to end of screen
- **1J**-Erase from start of screen to cursor
- **2J**-Erase entire screen

# terminal_codes.c

**A sample code showing the use of different terminal control codes**

# morev3.c

**Add a feature that displays a message in reverse video at the last line `--more--(?%)`**

# **morev4.c**

**Add a feature so that your more command can be used with input redirection symbol (<) as well as with the pipe symbol (|) on the shell**

# **morev5.c**

**Add a feature to compute and display %age of file contents displayed** `--more--(54%)`

# **morev6.c**

**Add the functionality that the user do not have to press enter after giving the command in the last line (non-canonical + non-echo mode)**

# morev7.c

**Add the functionality of computing the page size and displaying the file contents accordingly (even if the page size changes while more is running)**
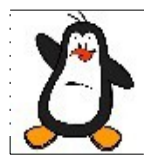
# morev8.c

**Add the functionality of command '/stringtosearch' for searching a string**

# morev9.c

**Add the functionality of command `'v'`, to open current file in `vim` and afterwards fall back**

# Things To Do



**If you have problems visit me in counseling hours. . . .**