“make” System

What to Expect?

- What is 'make' & its System?
- Learning 'make' by Examples

What is 'make'?

- Tool to automatically execute commands
  - Any number, Any length
- Based on complete dependency chains
  - Typically based on time (of change)
- Majorly replacing the manual process
  - Effectively saving time & manual errors
- Typical usages
  - Project build system (compilation, etc)
  - Packaging files (binaries, documentation, source code)
  - Installations (of Systems, Packages, ...)
  - Anything that can be automated based on dependencies

Usage of 'make'

- Initiated by the command 'make'
- Controlled by its configuration files
  - Containing rules to execute various commands
  - Based on various target dependencies
- Default configuration files
  - GNUmakefile, makefile, Makefile (in that order)
- Other configuration file using -f
- Assisted by many in-built rules & variables

Typical Rule

target: dependencies

<tab>action

<tab>action

<tab>...

Example

abc: pqr

    cp pqr abc

Sample Makefile

```
xyz: abc
    cp abc xyz

abc: pqr
    cp pqr abc

pqr:
    touch pqr
```

Let's create a Makefile
for
Automatically compiling a C program

Extending the Makefile

- Build an executable from many C files
- Build a project with 10,000 C files
- Add the user's header file dependencies
- Automatic header dependency generation

What all have we learnt?

- 'make' & its benefits
- Usage of 'make'
- Examples of using 'make'

Any Queries?