Shubham Singh / 2022700118

OS experiment 2 Static / dynamic linking report

Linking is a process in which externally defined objects are processed so as to make them operational.

The two types of linking are static and dynamic linking:
1) Static linking: The linker copies all library routines used in the program into executible image. This method requires more space but does not require prescence of library in the system while running, thus being faster and more portable.
2) Dynamic linking: Dynamic linking is performed during runtime. This linking is accomplished by placing the ~~na~~ name of a shareable library (i.e. a reference to the library) in the executible image. It requires less memory space as multiple programs can share a single copy of the library.

For static linking, we used the ~~ar~~ ar rcs command.
ar: archiver command in unix like systems
r: replace existing or new files
cz: create new archive if does not exist
s: Write an index or symbol ~~title~~ table

For dynamic linking we used the gcc -fPIC and export command:
gcc -fPIC: Compiles codes as position independent code, suitable for dynamic linking. -fPIC stands for position independent code.

export: It is used to make symbols defined in shared libraries accessible to other programs. for ex. $ export LD_LIBRARY_PATH = $path : $LD_LIBRARY_PATH sets the library search path.
(path to shared library)
The gcc -c command compiles a file without linking.