OS ASSIGNMENT 2

BY CHINTAN PUGGALOK
2019239
CSAM

THE COMMANDS USED IN MAKEFILE ARE AS FOLLOWS-

1.Pre processing

Command gcc -E add.c -o add.i
Syntax gcc -E input<.c> -o output<.i>
If we use just -E in gcc then it does only pre processing which includes

1. Removing comments- All the comments are removed from the program as they are written by humans for their own understanding and are of no use to the machine.
2. Expanding the macros - The macros defined by the programmer are placed at each call.
3. Expanding the included files- It includes the address of all parts of the libraries and files needed for compilation.

Analysis:

1. All data types have been provided a typedef.
2. Extern has been added for the stdio library.
3. Comments have been removed.

2. Compiling

Command gcc -S add.i -o add.s

Syntax gcc -S input<.i> -o output<.s>

This option compiles the code and produces an assembler file.

The code is converted in assembler instructions.

Analysis:

1. The externs have been removed.
2. Add and printf call appears in assembly code.
3. Registers are being used to move data.
4. Main has been declared as a global entry point.

3.1 Assembling

Command gcc -c add.s -o add2.o

Syntax gcc input<.s> -o output<.o>

The assembler converts the assembly code to machine language(object code).

Analysis:

1. The assembler has converted the assembly code into machine specific object code which is not human readable.

3.2 Assembling

nasm -felf64 add.asm -o add1.o

Syntax nasm -felf64 input<.asm> -o output<.o>

The asm assembler converts the assembly code to machine language code.

Analysis:

1. The assembler has converted the assembly code into machine specific object code which is not human readable.

3. Linking

gcc add1.o add2.o -o add

Syntax gcc input<.o> -o output<executable>

Now linking is done and there is addition of extra code to make the program run. Linker combines all the object code of libraries with the object code of our program and different files. The output is the executable file.

Analysis:

1. The object code from pre compiled libraries is linked with the object code of our program.
2. The object code from pre compiled files is linked with the object code of our program.
3. The output produced is an executable file which is system specific and has code for all the functions and libraries used.

Working of the program

1. After the compilation of both the object files.
2. The input is taken by the C program and the input goes to asm when the call appears.
3. The variables are put on stack and can also be accessed using rdi and rsi registers.
4. Then the value of rdi is added to rsi. Then rdi is moved to rax.
5. Then rax is returned and printf is called using the c program library.