MAKEFILE DOCUMENTATION

HARSH GANDHI

31 MARCH 2017

0.1 Introduction

A project may contain several component divisions. These components may have complex inter-dependencies.When a few changes are made to the source, manually recompiling the entire project each time is tedious, error-prone and time-consuming.

Make is a solution to these problems. It can be used to specify dependencies between components, so that it will compile components in the order required to satisfy dependencies. An important feature is that when a project is recompiled after a few changes, it will recompile only the files which are changed, and any components that are dependent on it. This saves a lot of time. Make is, therefore, an essential tool for a large software project.('Makefile' or 'makefile' are both acceptable.)

Makefile is a program building tool which runs on Unix, Linux, and their flavors. It aids in simplifying building program executables that may need various modules. To determine how the modules need to be compiled or recompiled together, make takes the help of user-defined makefiles. This tutorial should enhance your knowledge about the structure and utility of makefile.

0.2 Need for Makefile

Compiling the source code files can be tiring, especially when you have to include several source files and type the compiling command every time you need to compile. Makefiles are the solution to simplify this task.

Makefiles are special format files that help build and manage the projects automatically.

0.3 Sample Program

Performing a simple task of addition and subtraction of two numbers using multiple files.

File: header.h

```
#include < stdio.h >
extern int sum (int a, int b);
extern int dif (int a, int b);
```

File: main.c

```
#include "header.h"
int main ()
{
int a = 100, b=20;
sum (a,b);
dif (a,b);
return 0;
}
```

File: sum.c

```
#include "header.h"
int sum(int a, int b)
{
   printf ("Sum is = %d", a+b);
   return 0;
}
```

File: dif.c

```
#include "header.h"
int sum(int a, int b)
{

    int c = a-b;
    printf ("Difference is = %d", c);
    return 0;
}
```

File: makefile

```
harsh: sum.c dif.c main.c
gcc -o harsh sum.c dif.c main.c
```

0.4 Output

Executing the program in Terminal.

```
isea@isea-OptiPlex-7440-AIO: $ cd Desktop
isea@isea-OptiPlex-7440-AIO: /Desktop$ make
gcc -o harsh sum.c dif.c main.c
isea@isea-OptiPlex-7440-AIO: /Desktop$ ./harsh
Sum is = 120
Difference is = 80
isea@isea-OptiPlex-7440-AIO: /Desktop$
```