

## Make Utility

### Definition

**Header File:** A text file containing frequently used symbolic constants and macros. The header files end in .h by convention (e.g. stdio.h) and are listed in the program using #include.

**Macro:** certain textual pattern defined according to a set of rules. The interpreter or compiler automatically replaces the pattern when it is encountered.

**Source File:** A text file containing your C program.

### When we use make utility?

1. When you have a large program with many source and header files
2. The files depend on one another in complex ways

### Why we use make utility?

When you change a file with some dependent you have to recompile all of them.

### What is make utility?

Make utility automates the process of recompiling all necessary files

### How we use make utility?

Create makefile in the working directory

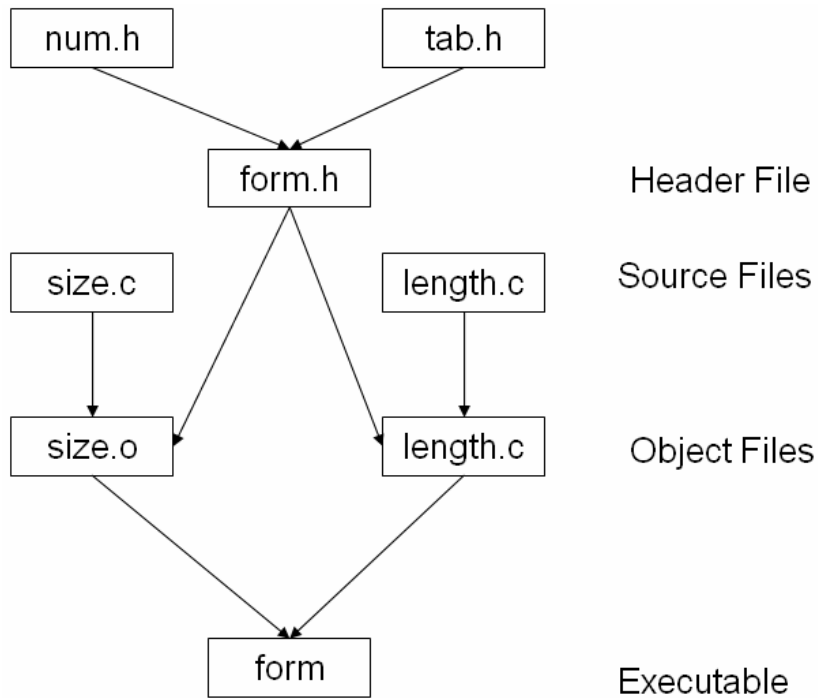
1. Define dependency lines
  - a. Dependency line indicated the relationship among files
  - b. Target: prerequisites list
2. Define construction commands
  - a. Unix shell commands
  - b. Starts with TAB
3. Execute makefile
  - a. Enter command make
4. Example

```
from:    size.o length.o
        cc -o from size.o length.o

size.o:  size.c form.h
        cc -c length.c

length.o:    length.c form.h
        cc -c length.c

form.h:  num.h tab.h
        cat num.h tab.h > form.h
```



## C Code

```
#num.h
#define NUM 5
```

```
#tab.h
#define TABSIZE 8
```

```
#length.c
#include <stdio.h>
#include "form.h"

void printLen()
{
    int len;
    len = TABSIZE;
    printf("Length is %d \n",len);
}
```

```
#size.c
#include <stdio.h>
#include "form.h"

void main()
{
    int size;
    size = NUM * TABSIZE;
    printLen();
    printf("Size is %d \n",size);
}
```

## Practice

