Data Structures

Makefile – Introduction

CODE
FORTHINGS

Makefile

Introduction

- Why Makefile?
- What is Makefile?
- Implementation

Introduction

Introduction

Introduction

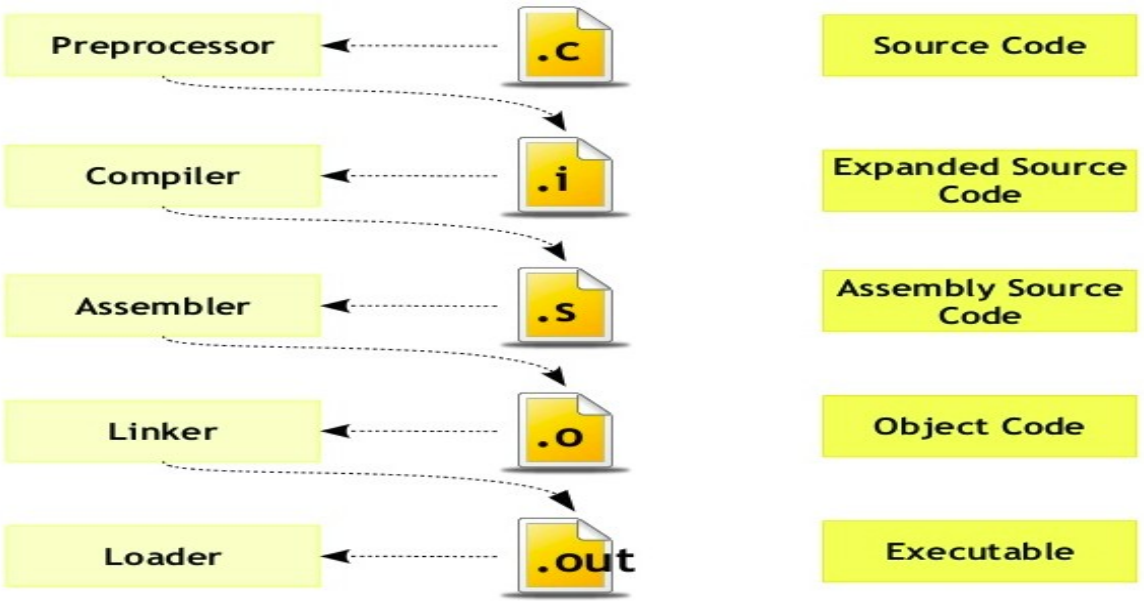
Different Stages of Compliation

- Preprocessor
- Compiler
- Assembler
- Linker
- Loader

Introduction

1.c ⟶ a.out

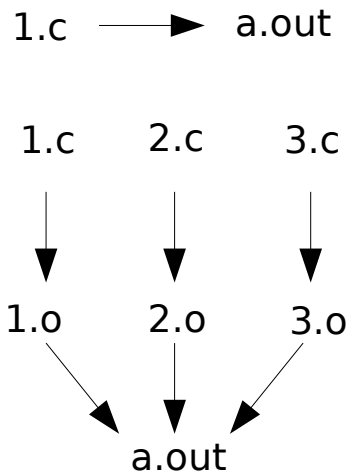
Compilation Stages



Introduction

1.c 2.c 3.c

Introduction



Introduction

Why?

Project ⟶ 1000 source file

Introduction

Why?

Project ⟶ 1000 source file

- gcc <all source files>
- gcc *.c

Introduction

What?

Makefiles are simple ways to organize code compilation

A makefile is a file containing a set of directives used by a make build automation tool to generate a target/goal.

Name

- Makefile
- makefile
- My_makefile make -f <MY_makefile>
- A makefile is executed using make command

Introduction

Rule

Target : Dependencies

<Tab> reciepe

Sub Target: Dependencies

<Tab> reciepe

clean:

<Tab>reciepe

Introduction

main.c add.c mul.c sub.c

Rule

Target : Dependencies

<Tab> reciepe

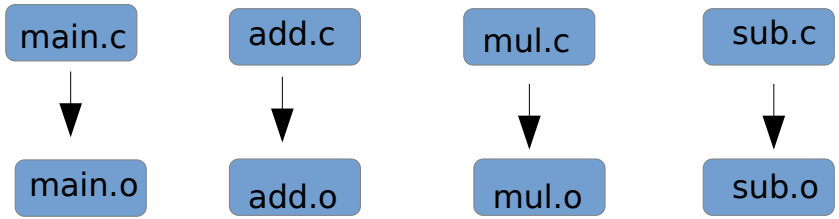
Sub Target: Dependencies

<Tab> reciepe

clean:

<Tab>reciepe

Introduction



Rule

Target : Dependencies

<Tab> reciepe

Sub Target: Dependencies

<Tab> reciepe

clean:

<Tab>reciepe

Introduction

Rule

Target : Dependencies

<Tab> reciepe

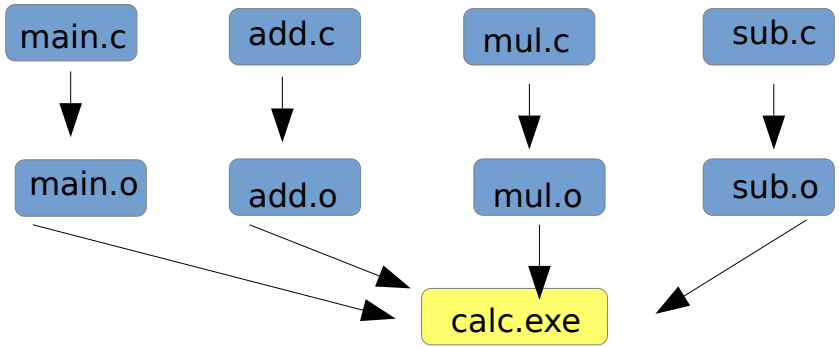
Sub Target: Dependencies

<Tab> reciepe

⋮

clean:

<Tab>reciepe



Introduction

Rule

Target : Dependencies

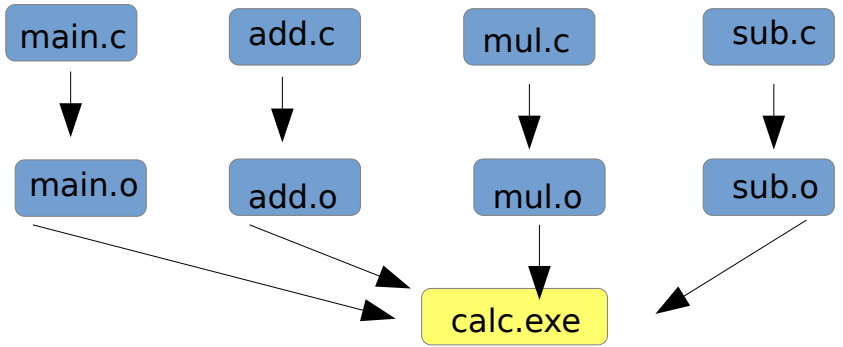
<Tab> reciepe

Sub Target: Dependencies

<Tab> reciepe

clean:

<Tab>reciepe




```
calc.exe : main.o add.o mul.o sub.o
<Tab>gcc -o calc.exe  main.o add.o mul.o sub.o
main.o: main.c
<Tab>gcc -c main.c
add.o: add.c
<Tab>gcc -c add.c
sub.o: sub.c
<Tab>gcc -c sub.c
clean:
<Tab>rm  *.exe  *.o
```

Introduction

Rule

Target : Dependencies

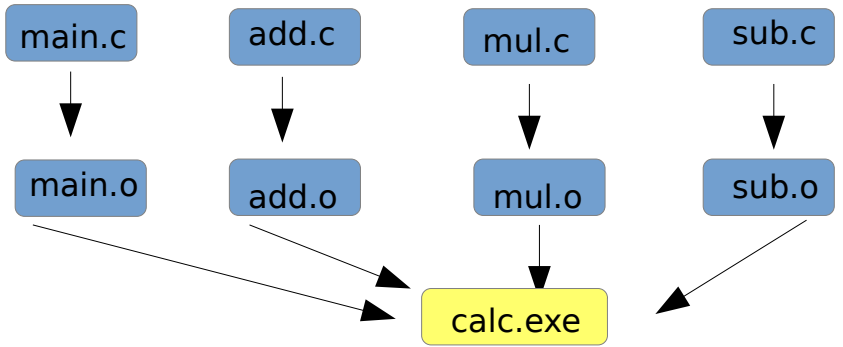
<Tab> reciepe

Sub Target: Dependencies

<Tab> reciepe

clean:

<Tab>reciepe




```
calc.exe : main.o add.o mul.o sub.o
<Tab>gcc -o calc.exe  main.o add.o mul.o sub.o
main.o: main.c
<Tab>gcc -c main.c
add.o: add.c
<Tab>gcc -c add.c
sub.o: sub.c
<Tab>gcc -c sub.c
clean:
<Tab>rm  *.exe  *.o
```

CFT CODE FORTHINGS

Makefile -Implementation

Introduction

Rule

Target : Dependencies

<Tab> reciepe

Sub Target: Dependencies

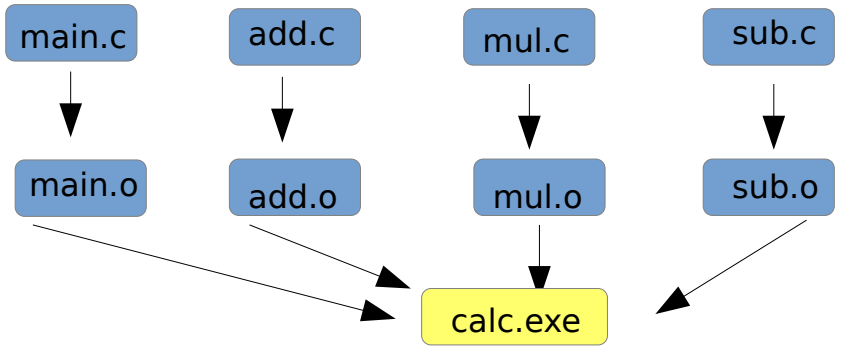
<Tab> reciepe

clean:

<Tab>reciepe

Alternate

```
OBJ : $(patsubst %.c, %.o, $(wildcard *.c))
calc.exe: $(OBJ)
<Tab> gcc -o  $@  $^
clean:
<Tab> rm  *.exe *.o
```




```
calc.exe : main.o add.o mul.o sub.o
<Tab>gcc -o calc.exe  main.o add.o mul.o sub.o
main.o: main.c
<Tab>gcc -c main.c
add.o: add.c
<Tab>gcc -c add.c
sub.o: sub.c
<Tab>gcc -c sub.c
clean:
<Tab>rm  *.exe  *.o
```

Introduction

Advantages

- The compliation is done using single command ie make
- Re compiles only what is needed when you make a change
- Make keeps track of time

Time Complexity