

A Minimal Book Example

Yihui Xie

2020-04-08

Contents

1	5
2 Makefile	7
2.1	7
2.2 make	8
2.3 makefile	9
2.4 make	10
2.5	10
2.6 Makefile	11
2.7	11
3 linux.sh	13
3.1	13
3.2	13
3.3	13
3.4	14
3.5 Shell	14
4 Methods	17
5 Applications	19
5.1 Example one	19
5.2 Example two	19
6 Final Words	21

Chapter 1

rbookdown

rbookdown PKU

Chapter 2

Makefile

```
Makefile CSDN ## Makefile      Makefile      Makefile
target... : prerequisites ...
        command
target      Object File      Label      “ ”      prerequisites      target      command      make
```

2.1

```
3      8 C      Makefile
edit : main.o kbd.o command.o display.o \
        insert.o search.o files.o utils.o
        cc -o edit main.o kbd.o command.o display.o \
                insert.o search.o files.o utils.o

main.o : main.c defs.h
        cc -c main.c

kbd.o : kbd.c defs.h command.h
        cc -c kbd.c
```

```

command.o : command.c defs.h command.h

        cc -c command.c

display.o : display.c defs.h buffer.h

        cc -c display.c

insert.o : insert.c defs.h buffer.h

        cc -c insert.c

search.o : search.c defs.h buffer.h

        cc -c search.c

files.o : files.c defs.h buffer.h command.h

        cc -c files.c

utils.o : utils.c defs.h

        cc -c utils.c

clean :

        rm edit main.o kbd.o command.o display.o \

        insert.o search.o files.o utils.o

        "Makefile" "makefile"          "make"      edit          "make
clean"      makefile      target      edit      *.o      prerequisites      .c
.h      .o      .o      edit

                                Tab      make          make targets prerequisites      prerequisites

        clean      C      lable      make          make      lable

```

2.2 make

```

        make      make      "Makefile" "makefile"          target      "edit"
edit      edit      .o      edit      edit      edit      .o      make      .o      .o
C H      make      .o      .o      make      edit      make      make
        clean          make      —"make clean"

                                file.c      file.o      file.o      file.o      edit      edit      e

```



```
"command.h" kdb.o command.o files.o      edit
```

2.3 makefile

```
edit

edit : main.o kbd.o command.o display.o \

        insert.o search.o files.o utils.o

cc -o edit main.o kbd.o command.o display.o \

        insert.o search.o files.o utils.o

[.o]          [.o]          clean      makefile      makefile

objects, OBJECTS, objs, OBJs, obj,  OBJ      obj      makefile

objects = main.o kbd.o command.o display.o \

        insert.o search.o files.o utils.o

makefile "$(objects)"          makefile

objects = main.o kbd.o command.o display.o \
        insert.o search.o files.o utils.o
edit : $(objects)
        cc -o edit $(objects)
main.o : main.c defs.h
        cc -c main.c
kbd.o : kbd.c defs.h command.h
        cc -c kbd.c
command.o : command.c defs.h command.h
        cc -c command.c
display.o : display.c defs.h buffer.h
        cc -c display.c
insert.o : insert.c defs.h buffer.h
        cc -c insert.c
search.o : search.c defs.h buffer.h
        cc -c search.c
files.o : files.c defs.h buffer.h command.h
        cc -c files.c
utils.o : utils.c defs.h
        cc -c utils.c
clean :
        rm edit $(objects)

.o          objects
```

2.4 make

```

GNU make                                [.o]                make

make [.o] [.c] make whatever.o whatever.c whatever.o
cc -c whatever.c makefile makefile

objects = main.o kbd.o command.o display.o \
          insert.o search.o files.o utils.o

edit : $(objects)
       cc -o edit $(objects)

main.o : defs.h
kbd.o : defs.h command.h
command.o : defs.h command.h
display.o : defs.h buffer.h
insert.o : defs.h buffer.h
search.o : defs.h buffer.h
files.o : defs.h buffer.h command.h
utils.o : defs.h

.PHONY : clean
clean :
       rm edit $(objects)

make " " ".PHONY" clean

```

2.5

```

Makefile .o " "

clean:

       rm edit $(objects)

.PHONY : clean

clean :

       -rm edit $(objects)

.PHONY clean " " rm clean make
—"clean"

```

makefile makefile makefile

2.6 Makefile

```

make            "GNUmakefile" "makefile" "Makefile"            "Makefile"            "GN
Makefile    "Make.Linux" "Make.Solaris" "Make.AIX"            Makefile    make "-
f" "-file"    make -f Make.Linux make -file Make.AIX

```

2.7

```

Makefile            C/C++            Makefile            C/C++            Makefile            Makefile            " " " " "
Makefile            " " " " " "            " " " " " "            "foo" "Foo" "FOO"            Makefile            Makefile
" "@

```

2.7.1

```

"$" ,    " "    "{"            "$"            "$$"            " " " " " " "
objects = program.o foo.o utils.o
program : $(objects)
cc -o program $(objects)

$(objects) : defs.h
          C/C++

foo = c
prog.o : prog.$(foo)
$(foo)$(foo) -$(foo) prog.$(foo)

prog.o : prog.c
cc -c prog.c

```

2.7.2

```

":="

x := foo
y := $(x) bar
x := later

y := foo bar
x := later

```

```
y := $(x) bar
x := foo
y "bar" "foo bar"
```

2.7.3

Makefile

- \$@
- \$%
- \$<
- \$?
- \$^
- \$+ “\$^”
- “ ”

Linux

.a

Makefile

.o

.o

\$*|

“\$”

“\$”

“ ” “ ”

“%”

1

test:test.o test1.o test2.o

gcc -o \$@ \$^

test.o:test.c test.h

gcc -o \$@ \$<

test1.o:test1.c test1.h

gcc -o \$@ \$<

test2.o:test2.c test2.h

gcc -o \$@ \$<

“\$@” “\$<” “\$^”

make make

Makefile

C

“\$@”

test “\$^”

“\$<”

2

lib:test.o test1.o test2.o

ar r \$?

make

“lib”

“\$?”

Chapter 3

linux.sh

3.1

```
#!/bin/sh
#
name ./filename /bin/sh chmod +x file-
```

3.2

```
shell #
```

3.3

```
shell
#!/bin/sh
#
a="hello world"
# a
echo "A is:"
echo $a

num=2
echo "this is the $numnd"
    "this is the 2nd"    "this is the " shell numnd shell num
```

```
num=2
echo "this is the ${num}nd"

    this is the 2nd
```

3.4

```
export
```

3.5 Shell

```
shell
```

3.5.1 1. Unix :

```
shell      unix      :
```

- echo "some text":
- ls:
- wc -l file wc -w file wc -c file:
- cp sourcefile destfile:
- mv oldname newname :
- rm file:
- grep 'pattern' file: grep 'searchstring' file.txt
- cut -b colnum file: 5 9 cut -b5-9 file.txt cat

- cat file.txt:
- file somefile:
- read var:
- sort file.txt: file.txt
- uniq: sort file.txt | uniq
- expr: Example: add 2 and 3 expr 2 "+" 3
- find: find . -name filename -print
- tee: () somecommand | tee outfile
- basename file: basename /bin/tux tux
- dirname file: dirname /bin/tux /bin
- head file:
- tail file :
- sed: Sed shell linuxfocus
- LinuxFocus

```
cat text.file | sed 's/linuxfocus/LinuxFocus/' >newtext.file
```

- awk: awk -F

```
cat file.txt | awk -F, '{print $1 "," $3}'
```

Adam Bor, 34, IndiaKerryMiller, 22, USA Adam
 Bor, IndiaKerry Miller, USA

3.5.2 2. : , backtick

```
(()) grep "hello" file.txt | wc -l file.txt "hello"
grep wc > >>
find . -mtime -1 -type f -print 24 -
mtime -2 48 linux #!/bin/sh # The ticks are
backticks (`) not normal quotes ('): tar -zcvf lastmod.tar.gz
`find . -mtime -1 -type f -print`
```

3.5.3 3.

```
1.if "if" then if ....; then .... elif ....; then .... else .... fi
... " [ ] " [ -f "somefile" ]
[ -x "/bin/ls" ] /bin/ls [ -n "$var" ] $var [ "$a" = "$b"
] $a $b man test \#!/bin/sh if [ "$SHELL"
= "/bin/bash" ]; then echo "your login shell is the bash (bourne
again shell)" else echo "your login shell is not bash but
$SHELL" fi #!/bin/sh mailfolder=/var/spool/mail/james [
-r "$mailfolder" ] { echo "Can not read $mailfolder" ; exit 1;
} echo "$mailfolder has mail from:" grep "^From " $mailfolder
mailfolder "From"
if
```

```
##
```

```
# # sac & td3
```

```
for exp in summon
```

```
do
```

```
# for version in pow-averse pow-seeking wang-averse wang-seeking neutral cpw-averse cpw-seeking
```

```
# for version in cpw-seeking cvar-seeking # pow-seeking wang-seeking
```

```
for version in neutral cpw-averse cvar-averse pow-averse wang-averse
```

```
do
```

```
for seed in 2
```

```
do
```

```
mkdir -p "logs/dsac-${exp}-${version}"
```

```
python dsac_v2.py --config "configs/highway/${exp}-${version}.yaml" --gpu 2 --seed ${s}
```

```
done
```

```
done
```

```
done
```


Chapter 4

Methods

We describe our methods in this chapter.

Chapter 5

Applications

Some *significant* applications are demonstrated in this chapter.

5.1 Example one

5.2 Example two

Chapter 6

Final Words

We have finished a nice book.