EE6094

CAD for VLSI Design

Checker and Makefile

Andy, Yu-Guang Chen
Assistant Professor, Department of EE
National Central University
andyygchen@ee.ncu.edu.tw

Slides Credit: 楊云緯

Outline

- ◆Checker
- ◆Makefile

Outline

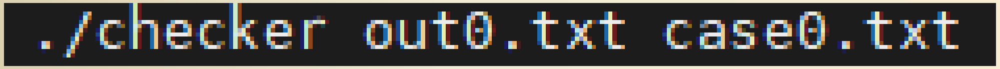
- ◆Checker
- ◆Makefile

Checker

- How to use the checker?
  - Step 1: Make sure you put your checker, testcase and your output file testcase_out.txt at the **same folder**.
  - Step 2: Make sure the format of your output file is correct.
  - Step 3: Key in the following commands.

    ./checker your_output_file your_input_file

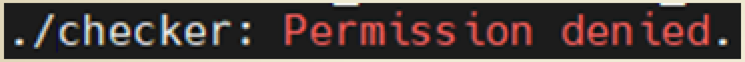
For example:



```
./checker out0.txt case0.txt
```

Checker

- If you do not have permission to execute checker:

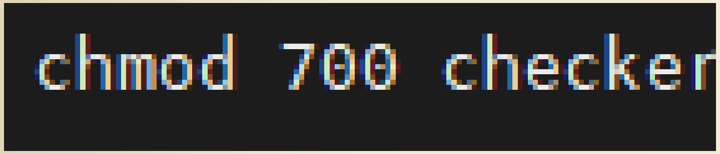


```
./checker: Permission denied.
```

  - Key in the following commands.

    chmod 700 checker

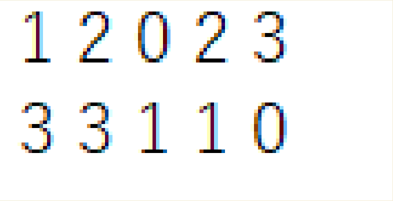
  For example:



```
chmod 700 checker
```

Checker

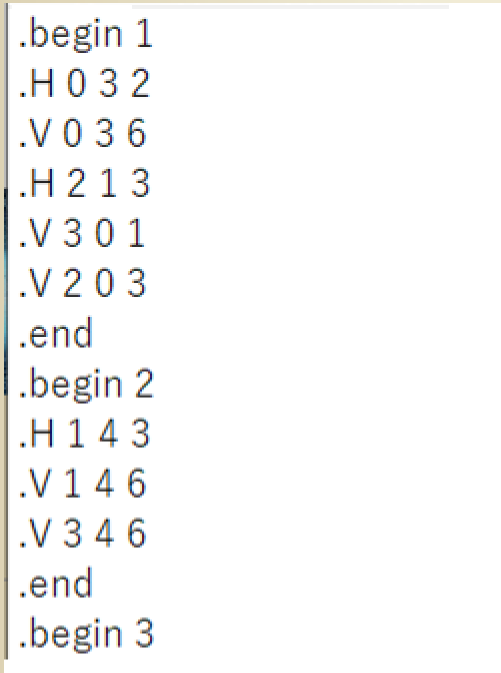
- Example case0.txt



```
1 2 0 2 3
3 3 1 1 0
```

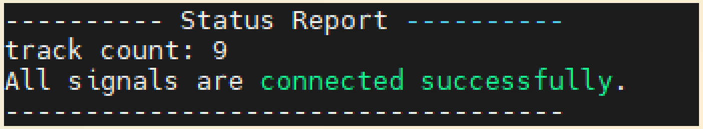
Checker

- If your program is correct, you will get “connected successfully” and the result of the track count.



```
.begin 1
.H 0 3 2
.V 0 3 6
.H 2 1 3
.V 3 0 1
.V 2 0 3
.end
.begin 2
.H 1 4 3
.V 1 4 6
.V 3 4 6
.end
.begin 3
```

testcase_out.txt

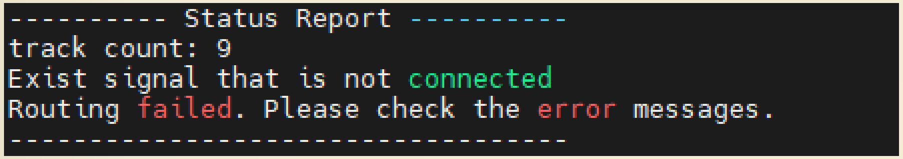


```
---------- Status Report ----------
track count: 9
All signals are connected successfully.
-----------------------------------
```

result

Checker

- If your program is wrong, you will get “failed” and the detail.



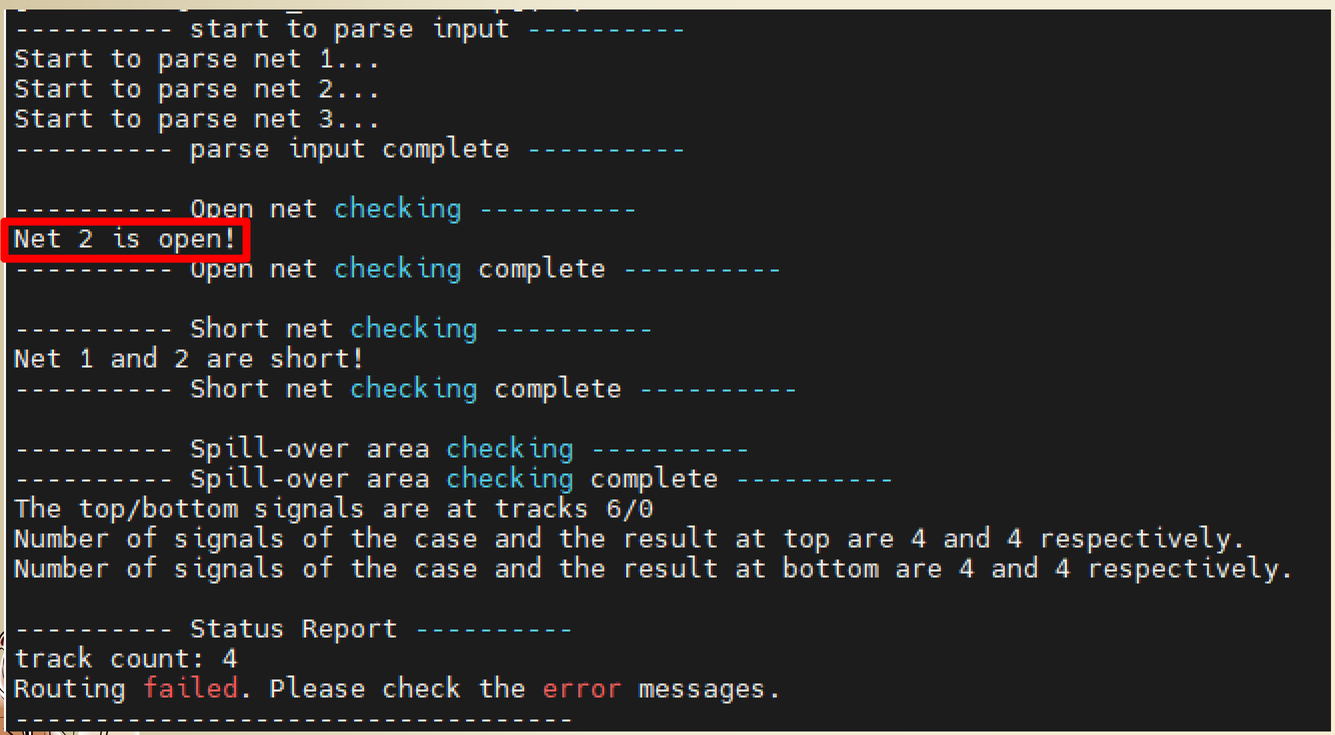
```
---------- Status Report ----------
track count: 9
Exist signal that is not connected
Routing failed. Please check the error messages.
-----------------------------------
```

Checker

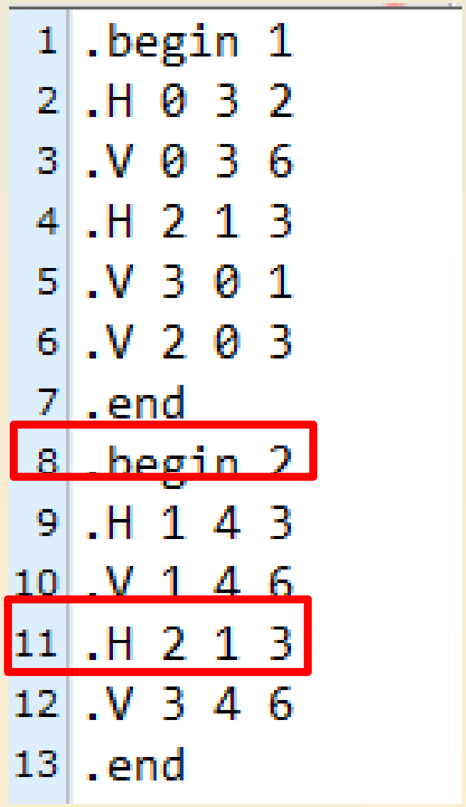
- Error type:
  - Open net
  - Short net
  - Spill-over area

Checker

- ◆ Open net
  - ➢ For example, the wire segment “H 2 1 3” in net2 is not connected.



```
---------- start to parse input ----------
Start to parse net 1...
Start to parse net 2...
Start to parse net 3...
---------- parse input complete ----------

---------- Open net checking ----------
Net 2 is open!
---------- Open net checking complete ----------

---------- Short net checking ----------
Net 1 and 2 are short!
---------- Short net checking complete ----------

---------- Spill-over area checking ----------
---------- Spill-over area checking complete ----------
The top/bottom signals are at tracks 6/0
Number of signals of the case and the result at top are 4 and 4 respectively.
Number of signals of the case and the result at bottom are 4 and 4 respectively.

---------- Status Report ----------
track count: 4
Routing failed. Please check the error messages.
-----------------------------------
```

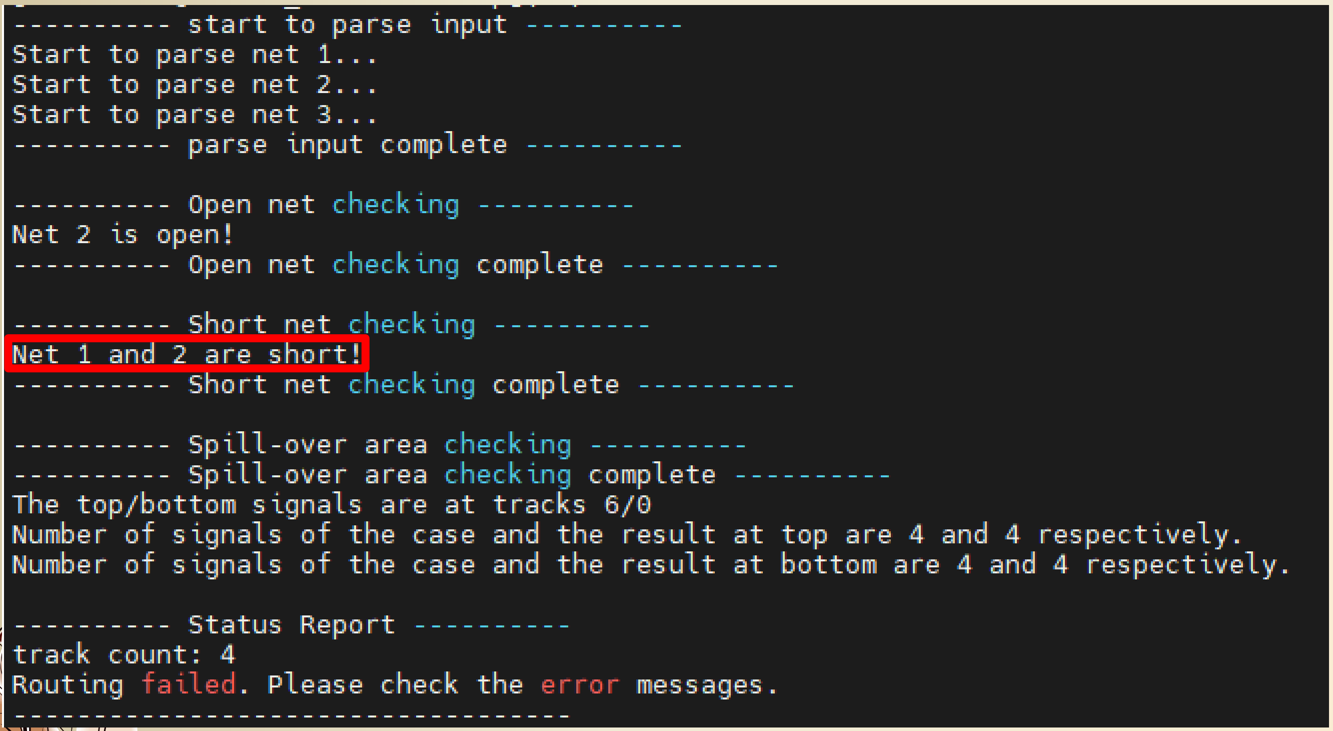


```
.begin 1
.H 0 3 2
.V 0 3 6
.H 2 1 3
.V 3 0 1
.V 2 0 3
.end
.begin 2
.H 1 4 3
.V 1 4 6
.H 2 1 3
.V 3 4 6
.end
```

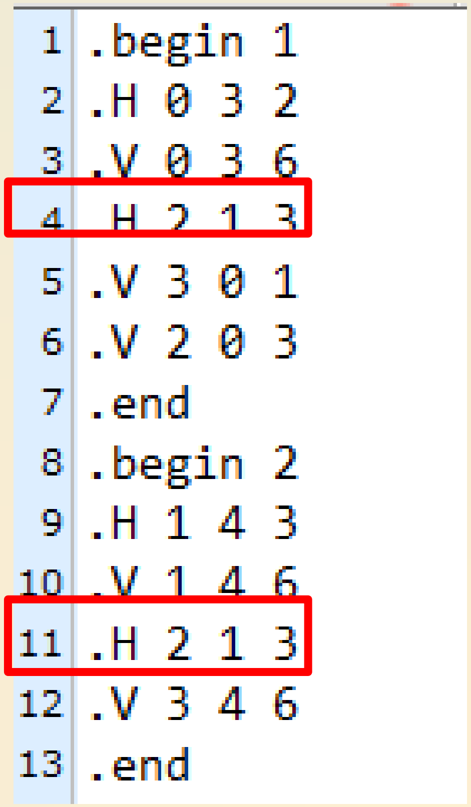
Checker

◆Short net

- ➢For example, the wire segment “H 2 1 3” appears in net2 and net1 at the same time.



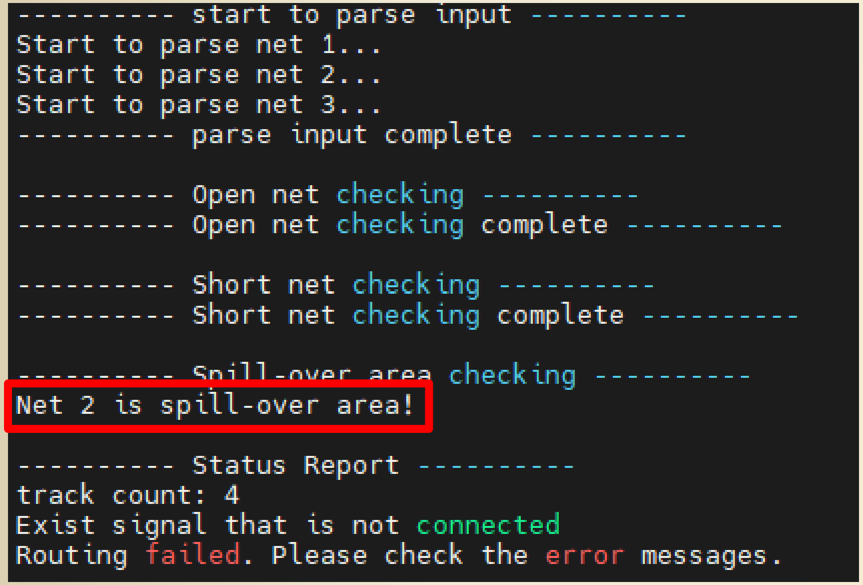
```
---------- start to parse input ----------
Start to parse net 1...
Start to parse net 2...
Start to parse net 3...
---------- parse input complete ----------

---------- Open net checking ----------
Net 2 is open!
---------- Open net checking complete ----------

---------- Short net checking ----------
Net 1 and 2 are short!
---------- Short net checking complete ----------

---------- Spill-over area checking ----------
---------- Spill-over area checking complete ----------
The top/bottom signals are at tracks 6/0
Number of signals of the case and the result at top are 4 and 4 respectively.
Number of signals of the case and the result at bottom are 4 and 4 respectively.

---------- Status Report ----------
track count: 4
Routing failed. Please check the error messages.
-----------------------------------
```



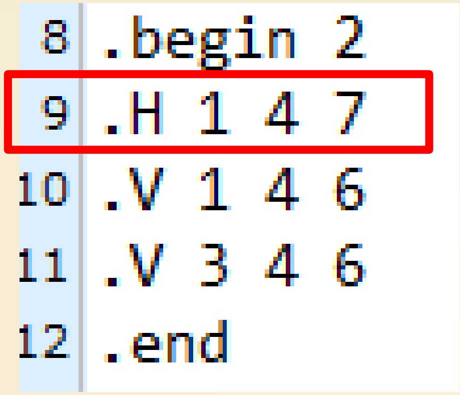
```
.begin 1
.H 0 3 2
.V 0 3 6
.H 2 1 3
.V 3 0 1
.V 2 0 3
.end
.begin 2
.H 1 4 3
.V 1 4 6
.H 2 1 3
.V 3 4 6
.end
```

Checker

- Spill-over area
  - Take case0.txt for example, the wire segment “H 1 4 7” exceed the #column which is 5 here.



```
---------- start to parse input ----------
Start to parse net 1...
Start to parse net 2...
Start to parse net 3...
---------- parse input complete ----------

---------- Open net checking ----------
---------- Open net checking complete ----------

---------- Short net checking ----------
---------- Short net checking complete ----------

---------- Spill-over area checking ----------
Net 2 is spill-over area!

---------- Status Report ----------
track count: 4
Exist signal that is not connected
Routing failed. Please check the error messages.
```



```
.begin 2
.H 1 4 7
.V 1 4 6
.V 3 4 6
.end
```

Outline

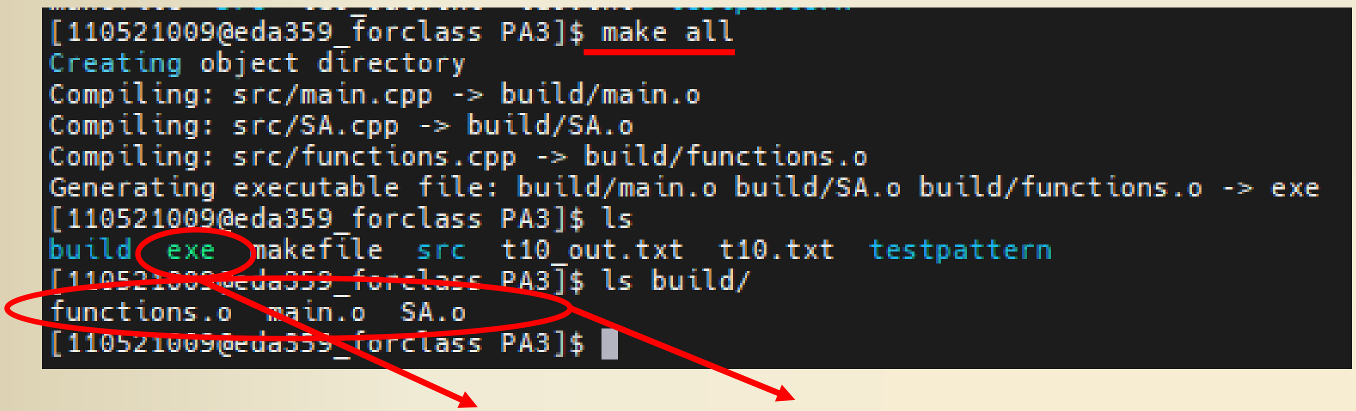
- Checker
- Makefile

Makefile

- In demo session, TA will test your makefile in the following commands.
  - *make all*
  - *make run input=your_input_file output=your_output_file*
  - *make clean*

Makefile

- *make all*

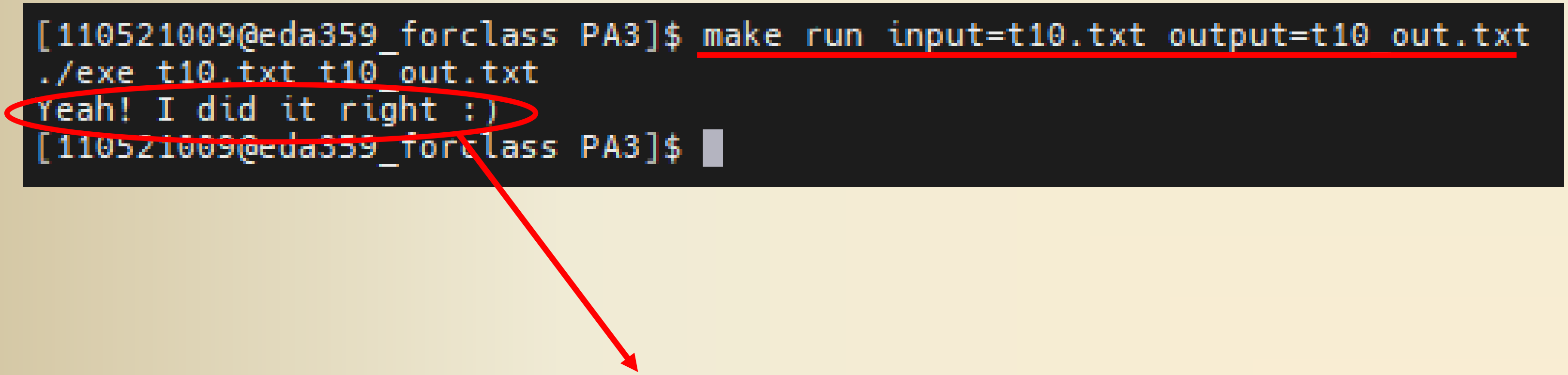


```
[110521009@eda359_forclass PA3]$ make all
Creating object directory
Compiling: src/main.cpp -> build/main.o
Compiling: src/SA.cpp -> build/SA.o
Compiling: src/functions.cpp -> build/functions.o
Generating executable file: build/main.o build/SA.o build/functions.o -> exe
[110521009@eda359_forclass PA3]$ ls
build  exe  makefile  src  t10_out.txt  t10.txt  testpattern
[110521009@eda359_forclass PA3]$ ls build/
functions.o  main.o  SA.o
[110521009@eda359_forclass PA3]$
```

After *make all* , an executable file and object file are generated

Makefile

- *make run*

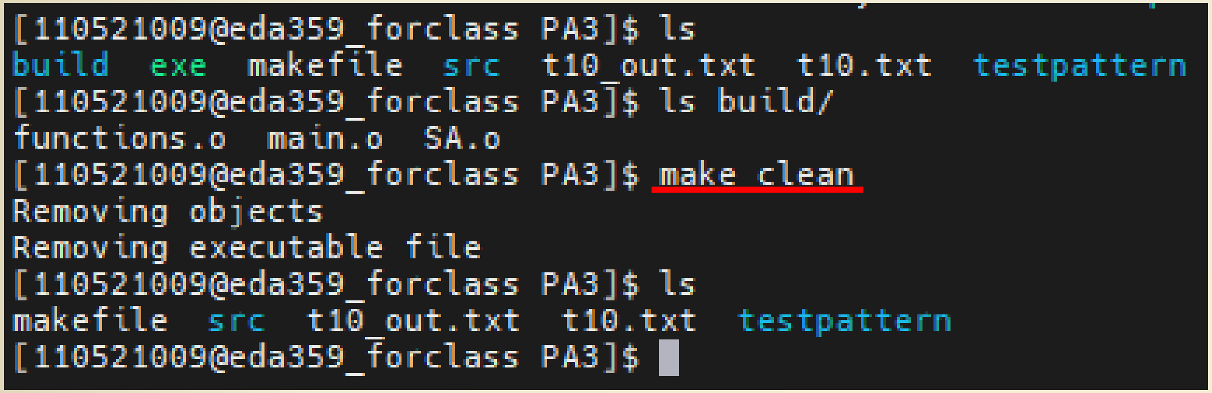


```
[110521009@eda359_forclass PA3]$ make run input=t10.txt output=t10_out.txt
./exe t10.txt t10_out.txt
Yeah! I did it right :)
[110521009@eda359_forclass PA3]$ 
```

Note that using cout will slow down runtime.

Makefile

- *make clean*



```
[110521009@eda359_forclass PA3]$ ls
build  exe  makefile  src  t10_out.txt  t10.txt  testpattern
[110521009@eda359_forclass PA3]$ ls build/
functions.o  main.o  SA.o
[110521009@eda359_forclass PA3]$ make clean
Removing objects
Removing executable file
[110521009@eda359_forclass PA3]$ ls
makefile  src  t10_out.txt  t10.txt  testpattern
[110521009@eda359_forclass PA3]$ 
```