Lab0实验报告

姓名：吴进兴

学号：519021910919

```
❯ unzip lab0.zip
Archive:  lab0.zip
   creating: lab0/
  inflating: lab0/ll_equal.c
  inflating: lab0/hello.c
  inflating: lab0/glory.c
  inflating: lab0/wc.c

 ~/Downloads                                                    20:45:18
❯
```

##练习1 gcc

```
❯ cd lab0
❯ ls
glory.c  hello.c  ll_equal.c  wc.c
❯ gcc -o glory glory.c
zsh: command not found: gcc
```

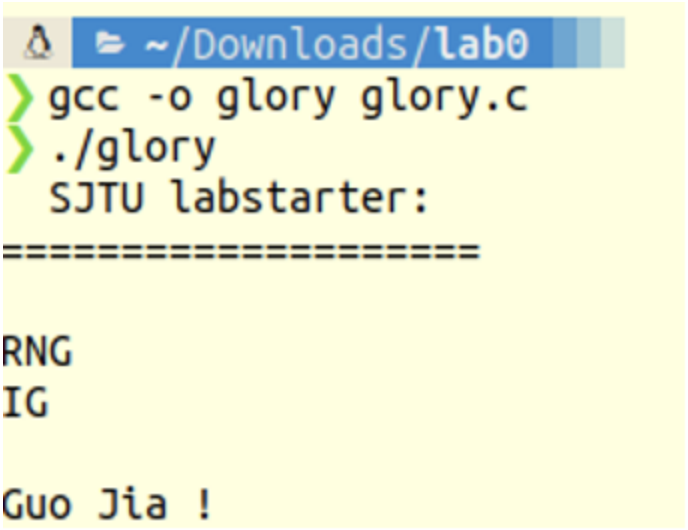
发现gcc并未安装。因此先安装gcc工具。

```
❯ sudo apt update
[sudo] password for frodo:
```

```
❯ sudo apt install build-essential
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

```
❯ gcc --version
gcc (Ubuntu 9.4.0-1ubuntu1~20.04) 9.4.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

安装gcc9.4.0版本。进而练习gcc操作。



##练习2 GDB

```
❯ gcc -g -o hello hello.c
❯ gdb hello
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from hello...
(gdb)
```

1. How do you pass command line arguments to a program when using gdb?

run+程序参数/gdb --args xxx 参数

2. How do you set a breakpoint which only occurs when a set of conditions is true (e.g. when certain variables are a certain value)?

在条件跳转到的位置设置断点

3. How do you execute the next line of C code in the program after stopping at a breakpoint?

next/n

4. If the next line of code is a function call, you'll execute the whole function call at once if you use your answer to #3. How do you tell GDB that you want to debug the code inside the function instead?

step

5. How do you resume the program after stopping at a breakpoint?

continue/c

6. How can you see the value of a variable (or even an expression like 1+2) in gdb?

print a/p a (a is the variable which you want to check)

If want print a expression print 1+2

7. How do you configure gdb so it prints the value of a variable after every step?

display a (a is the variable which you want to check)

8. How do you print a list of all variables and their values in the current function?

info locals

9. How do you exit out of gdb?

quit/q

##练习3 调试

```
/* FIXME: this function is buggy. */
int ll_equal(const node* a, const node* b) {
  while (a != NULL&& b != NULL) {
    if (a->val != b->val)
      return 0;
    a = a->next;
    b = b->next;
  }
  /* lists are equal if a and b are both null */
  return a == b;
}
```

##练习4 Make初步

```
void wc(FILE *ofile, FILE *infile, char *inname) {
  int characters, lines, words, state;
  char c;
  state = characters = lines = words = 0;
  if(inname == NULL){

  while((c = getchar()) != '0') {
        characters++;
        if(c == '\n') {
            lines++;
            state = 0;
            continue;
        } else if(c == ' ') {
```

```
            state = 0;
            continue;
        } else if(c == '\t') {
            state = 0;
            continue;
        } else {
            if(state == 0) {
                state = 1;
                words++;
            }
            continue;
        }
    }
}

  else if(inname != NULL && out_file == NULL){

  fp=fopen(inname,"r");

    while((c=fgetc(fp)) != '0') {
        characters++;
        if(c == '\n') {
            lines++;
            state = 0;
            continue;
        } else if(c == ' ') {
            state = 0;
            continue;
        } else if(c == '\t') {
            state = 0;
            continue;
        } else {
            if(state == 0) {
                state = 1;
                words++;
            }
            continue;
        }
    }
  fclose(fp);
}

   else (inname != NULL && out_file != NULL){

  fp=fopen(inname,"r");

    while((c=fgetc(fp)) != '0') {
        characters++;
        if(c == '\n') {
            lines++;
            state = 0;
            continue;
        } else if(c == ' ') {
            state = 0;
```

```
            continue;
        } else if(c == '\t') {
            state = 0;
            continue;
        } else {
            if(state == 0) {
                state = 1;
                words++;
            }
            continue;
        }
    }
  fclose(fp);
  fp=fopen(out_file,"w+");
  fprintf(fp, "%d characters. %d words. %d lines.\n", characters, words, lines);
  fclose(fp);

}

    printf("%d characters. %d words. %d lines.\n", characters, words, lines);

  return (0);

}
```