

lab0 report

520021911347 郑航

exercise2

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

Use the set args command to set the command line parameters before using the run command.

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)?

Use the break ... if cond command, and ... is the parameter to settle where the breakpoint is, cond is an expression. Only when the cond expression is true will the program stop at the breakpoint.

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

There are two ways. The first one is to use the next command (abbreviated as n), while the second one is to use the step command (abbreviated as s). The difference of these two ways is that when occurring to a function, the next command will take the function as a single line and skip it at one step while the step command will get inside the function to debug inside.

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?

As in the answer to #3, we can use the step command instead of the next command.

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

Use the continue command (abbreviated as c).

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

Use the print command (abbreviated as p).

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

Use the display command (abbreviated as disp).

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

Use the info variables command.

9. How do you exit out of gdb?

Use the quit command (abbreviated as q).

exercise3

The modified codes are as below:

```
1 #include <stdio.h>
2
```

```

3  typedef struct node {
4      int val;
5      struct node* next;
6  } node;
7
8  /* FIXME: this function is buggy. */
9  int ll_equal(const node* a, const node* b) {
10     while (a != NULL && b != NULL ) {
11         if (a->val != b->val)
12             return 0;
13         a = a->next;
14         b = b->next;
15     }
16     /* lists are equal if a and b are both null */
17     return a == b;
18 }
19
20 int main(int argc, char** argv) {
21     int i;
22     node nodes[10];
23
24     for (i=0; i<10; i++) {
25         nodes[i].val = 0;
26         nodes[i].next = NULL;
27     }
28
29     nodes[0].next = &nodes[1];
30     nodes[1].next = &nodes[2];
31     nodes[2].next = &nodes[3];
32
33     printf("equal test 1 result = %d\n", ll_equal(&nodes[0], &nodes[0]));
34     printf("equal test 2 result = %d\n", ll_equal(&nodes[0], &nodes[2]));
35
36     return 0;
37 }
38

```

exercise4

Function wc in wc.c:

```

1  void wc(FILE *ofile, FILE *infile, char *iname) {
2      // your code here:
3      char result[20];
4      int word=0, count=0, line=0;
5      if(!iname) iname="";
6      if(!infile){
7          char ch;
8          int isblank=0;
9          int isfirst=1;
10         while(scanf("%c",&ch)!=EOF){
11             if(isfirst)if(ch!=' '){word++;isfirst=0;}
12             if(ch=='\n'){line++;word++;}
13             else if(ch==' '){if(!isblank){word++;isblank=1;}
14             else isblank=0;

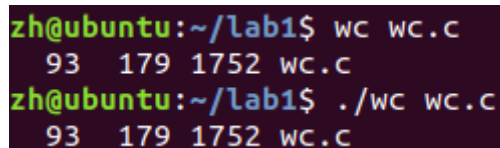
```

```

15         count++;
16     }
17 }
18 else{
19     char str[1000];
20     while(!feof(infile)){
21         fgets(str,1000,infile);
22         line++;
23         int i=0;
24         for(i=0;str[i]!='\n';i++)count++;
25         if(str[i]=='\n')count++;
26         char *tmp2;
27         char* tmp=str;
28         while (*tmp !='\0') {
29             if(!((*tmp>=33)&&(*tmp<=126))) {
30                 tmp++;
31                 continue;
32             }
33             else {
34                 tmp2=tmp+1;
35                 do {
36                     if(!((*tmp2>=33)&&(*tmp2<=126))) {
37                         word++;
38                         tmp=tmp2;
39                         break;
40                     }
41                 } while (*(++tmp2)!= '\0');
42             }
43         }
44     }
45 }
46 line--;
47 count-=2;
48 word--;
49 }
50 if(!ofile)printf(" %d %d %d %s\n",line,word,count,inname);
51 else fprintf(ofile," %d %d %d %s\n",line,word,count,inname);
52 }
53

```

结果截图如下:



```

zh@ubuntu:~/lab1$ wc WC.C
93 179 1752 WC.C
zh@ubuntu:~/lab1$ ./WC WC.C
93 179 1752 WC.C

```