

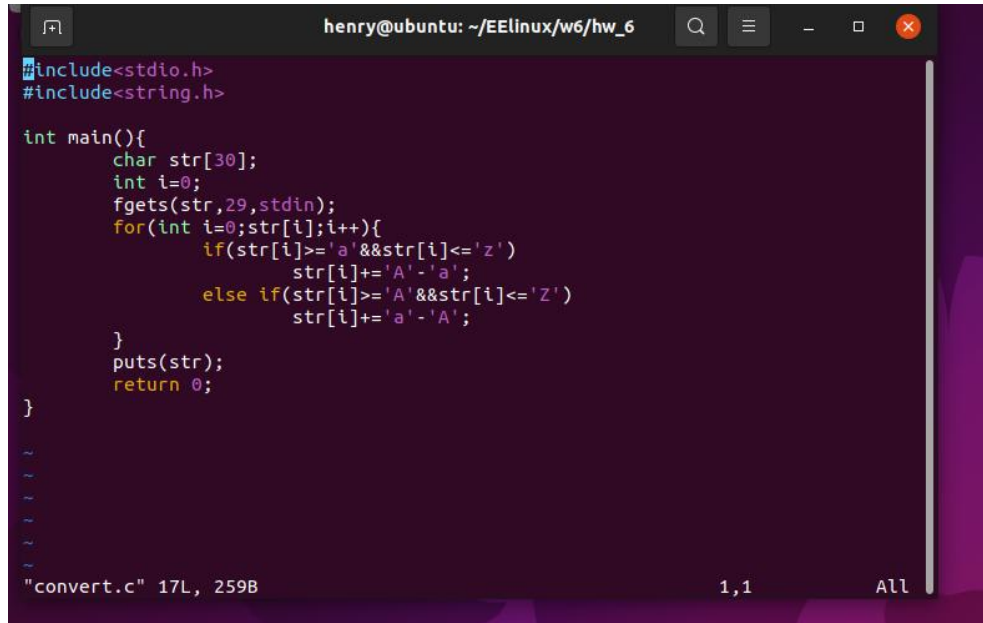
# Linux 第六周作业

201180189 周子航

1. 编写一个将输入的字符串进行大小写转换的程序。具体要求：

(1) 输入字符串包括大小写英文字母和阿拉伯数字，将大写字母转换成小写字母，小写字母转换成大写字母，数字不变，然后输出到屏幕上。例如，输入是 A89bcD，输出则为 a89BCd。

1.编写 C 程序

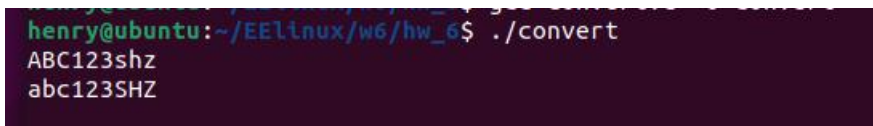


```
henry@ubuntu: ~/EElinux/w6/hw_6
#include<stdio.h>
#include<string.h>

int main(){
    char str[30];
    int i=0;
    fgets(str,29,stdin);
    for(int i=0;str[i];i++){
        if(str[i]>='a'&&str[i]<='z')
            str[i]+='A'-'a';
        else if(str[i]>='A'&&str[i]<='Z')
            str[i]+='a'-'A';
    }
    puts(str);
    return 0;
}

"convert.c" 17L, 259B 1,1 All
```

2. 运行结果



```
henry@ubuntu: ~/EElinux/w6/hw_6$ ./convert
ABC123shz
abc123SHZ
```

(2) 通过 `#ifdef` `#endif` 的方式定义 `CONVERT` 宏，将上述转换输出的代码保护起来。实现判断，如果定义 `CONVERT` 宏，则输出转换后的字符串结果，如果没有定义 `CONVERT` 宏，则仅输出自己的学号。

#### 1. 添加条件编译条件

```
#ifdef CONVERT

#include<stdio.h>
#include<string.h>

int main(){
    char str[30];
    int i=0;
    fgets(str,29,stdin);
    for(int i=0;str[i];i++){
        if(str[i]>='a'&&str[i]<='z')
            str[i]+='A'-'a';
        else if(str[i]>='A'&&str[i]<='Z')
            str[i]+='a'-'A';
    }
    puts(str);
    return 0;
}

#else

#include<stdio.h>
int main(){
    printf("201180189\n");
    return 0;
}

#endif
```

3. 使用 `gcc` 命令，给出合适的选项和参数进行如上的编译和链接，要求两种输出结果都有截图。

#### 1. 第一种情况，`CONVERT` 宏有定义

```
henry@ubuntu:~/EElinux/w6/hw_6$ gcc -o convert1 convert.c -D CONVERT
henry@ubuntu:~/EElinux/w6/hw_6$ ./convert1
ABC123def
abc123DEF
```

#### 2. 第二种情况，`CONVERT` 宏无定义

```
henry@ubuntu:~/EElinux/w6/hw_6$ gcc -o convert2 convert.c
henry@ubuntu:~/EElinux/w6/hw_6$ ./convert2
201180189
henry@ubuntu:~/EElinux/w6/hw_6$
```

2. 编写一个 Makefile，包含三个目标（输出转换结果、输出未转换结果，clean 目标用于清除源文件以外的中间文件和输出文件），使用 make 命令执行三个目标相关的命令，有截图。

1. 编写 makefile 文件

```
src=convert.c
target1=convert1
target2=convert2

$(target1):$(src)
    gcc $(src) -o $(target1) -D CONVERT
    ./$(target1)
$(target2):$(src)
    gcc $(src) -o $(target2)
    ./$(target2)

.PHONY:clean
clean:
    rm $(target1) $(target2)
```

3. 测试三种运行结果

```
henry@ubuntu:~/EElinux/w6/hw_6$ ls
convert.c  makefile
henry@ubuntu:~/EElinux/w6/hw_6$ vim makefile
henry@ubuntu:~/EElinux/w6/hw_6$ make convert1
gcc convert.c -o convert1 -D CONVERT
./convert1
123ABCdef456
123abcDEF456

henry@ubuntu:~/EElinux/w6/hw_6$ make convert2
gcc convert.c -o convert2
./convert2
201180189

henry@ubuntu:~/EElinux/w6/hw_6$ ls
convert1  convert2  convert.c  makefile
henry@ubuntu:~/EElinux/w6/hw_6$ make clean
rm convert1 convert2
henry@ubuntu:~/EElinux/w6/hw_6$ ls
convert.c  makefile
henry@ubuntu:~/EElinux/w6/hw_6$
```

4. 使用 gdb 命令, 调试上述编译链接生成的可执行程序, 要求对 gdb 常用的命令进行熟悉, 有截图。

1.没有输出结果的程序进行调试

```
<https://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 app...
(gdb) l
10         for(int i=0;str[i];i++){
11             if(str[i]>='a'&&str[i]<='z')
12                 str[i]+='A'-'a';
13             else if(str[i]>='A'&&str[i]<='Z')
14                 str[i]+='a'-'A';
15         }
16         puts(str);
17         return 0;
18     }
19
(gdb) l
20 #else
21
22 #include<stdio.h>
23 int main(){
24     printf("201180189\n");
25     return 0;
26 }
27
28 #endif
(gdb) n
The program is not being run.
(gdb) s
The program is not being run.
(gdb) start
Temporary breakpoint 1 at 0x1151: file convert.c, line 24.
Starting program: /home/henry/EELinux/w6/hw_6/app
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

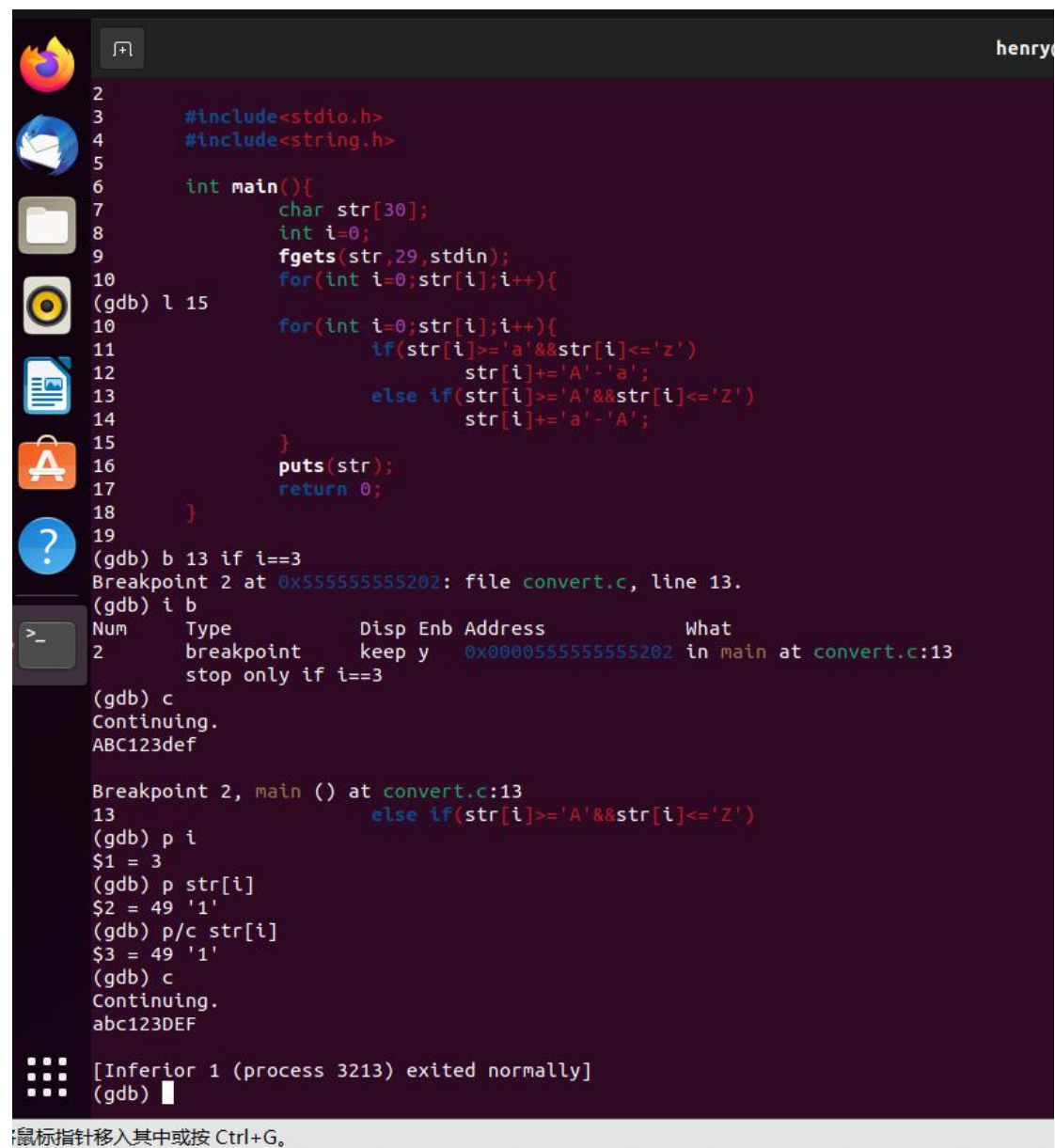
Temporary breakpoint 1, main () at convert.c:24
24     printf("201180189\n");
(gdb) n
201180189
25     return 0;
(gdb) █
```

1.l 用于显示代码

2.start 启动程序

3.n 单步调试

## 2. 有输出结果的程序进行调试



```
2
3     #include<stdio.h>
4     #include<string.h>
5
6     int main(){
7         char str[30];
8         int i=0;
9         fgets(str,29,stdin);
10        for(int i=0;str[i];i++){
11
12            for(int i=0;str[i];i++){
13                if(str[i]>='a'&&str[i]<='z')
14                    str[i]+='A'-'a';
15                else if(str[i]>='A'&&str[i]<='Z')
16                    str[i]+='a'-'A';
17            }
18        }
19        puts(str);
20        return 0;
21    }
22
23 (gdb) b 13 if i==3
24 Breakpoint 2 at 0x55555555202: file convert.c, line 13.
25 (gdb) i b
26
27   Num      Type             Disp Enb Address            What
28   --      -
29   2        breakpoint       keep y   0x000055555555202 in main at convert.c:13
30                                     stop only if i==3
31
32 (gdb) c
33 Continuing.
34 ABC123def
35
36 Breakpoint 2, main () at convert.c:13
37 13                                     else if(str[i]>='A'&&str[i]<='Z')
38 (gdb) p i
39 $1 = 3
40 (gdb) p str[i]
41 $2 = 49 '1'
42 (gdb) p/c str[i]
43 $3 = 49 '1'
44 (gdb) c
45 Continuing.
46 abc123DEF
47
48 [Inferior 1 (process 3213) exited normally]
49 (gdb)
```

鼠标指针移入其中或按 Ctrl+G。

1. *b* 设置断点
2. *p* 打印对应变量信息
3. *i* 查看断点信息