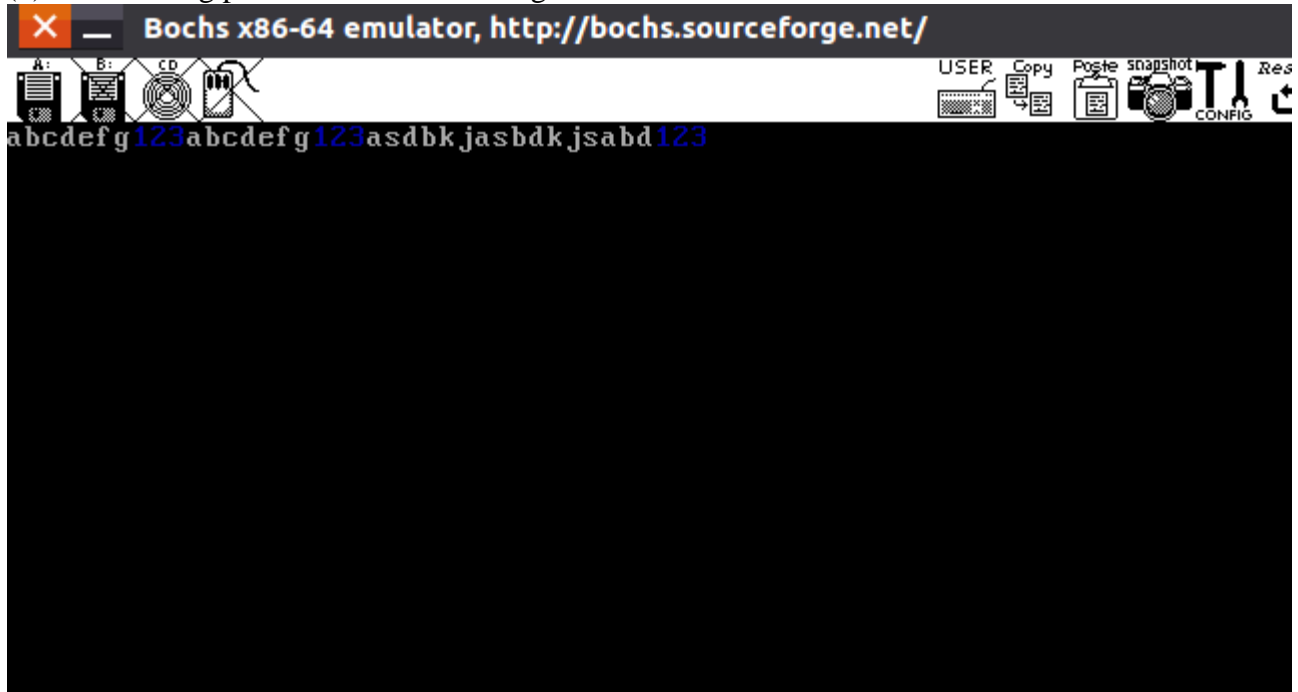


说明文档: 王瑞华 161250143

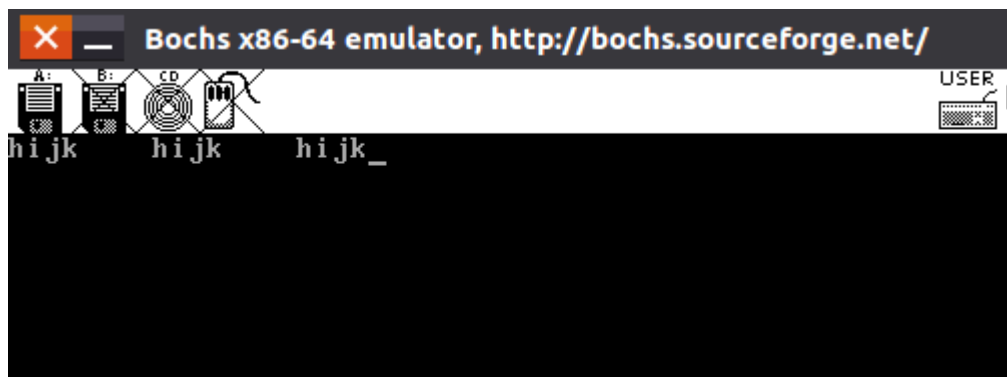
(1) in this version, I have already finished all the function required in the document by myself, the code is based on Orange's chapter 7 , and I added about 200 lines of code.

(2)go into the root position, open terminal and input 'make run', then the bochs will start.

(3)the following picture shows the finding function:



(4)the following picture show the TAB function:



(5) cleaning-screen is also done, you can try it by yourself.

(6) the following pictures show some of my additions

proto.h × global.h × keymap.h × console.c × tty.c × const.h ×

CMake project is not loaded

```
10 #include "protect.h"
11 #include "string.h"
12 #include "proc.h"
13 #include "tty.h"
14 #include "console.h"
15 #include "global.h"
16 #include "keyboard.h"
17 #include "proto.h"
18
19 PRIVATE void set_cursor(unsigned int position);
20 PRIVATE void set_video_start_addr(u32 addr);
21 PRIVATE void flush(CONSOLE* p_con);
22
23 extern int isFindState;
24 //这里是我新加的函数和数据结构，为了实现Tab的后退
25 int isTab(int pos); //判断当前光标位置是不是Tab键，1 是，0 不是
26 int TabPos[128]; //记录是Tab 位置的光标
27 int TabPtr=-1; //记录TabPos数组的位置
28 void pushTab(int pos); //Tab 位置压栈
29 void popTab(); //Tab 位置出栈
30 void cleanTabStack(); // Tab 栈清空 (这个方法按照当前实现是不需要使用的)
31
32 //这里是我新加的函数和数据结构，为了实现回车的后退
33 int isEnter(int pos); //判断当前光标位置是不是回车键，>0 是，0 不是。返回delta
34 int EnterPos[128]; //记录回车位置的光标
35 int delta[128]; //记录回车位置光标与上一字符光标的差值
36 int EnterPtr=-1; //记录回车数组的位置
37 void pushEnter(int pos,int delta);
38 void popEnter();
39
40 //用于记录查找内容的数据结构
41 int findlen=0;
42
43 /*=====*
```

CMake project is not loaded

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

//下面的代码和方法是我自己加的，为了实现Tab和回车的整体后退

void pushTab(int pos){

TabPtr++;

TabPos[TabPtr]=pos;

}

void popTab(){

if(TabPtr>=0)

TabPtr--;

}

void cleanTabStack(){

TabPtr=-1;

}

int isTab(int pos){

if(TabPos[TabPtr]==pos){

popTab();

return 1;

}else{

return 0;

}

}

void pushEnter(int pos,int delta0){

EnterPtr++;

EnterPos[EnterPtr]=pos;

delta[EnterPtr]=delta0;

}

void popEnter(){

if(EnterPtr>=0)

EnterPtr--;

}

int isEnter(int pos){

int ret=0;

if(EnterPos[EnterPtr]==pos){

ret=delta[EnterPtr];

popEnter();

}

return ret;

}

PUBLIC void stop\_find(CONSOLE\* p\_con){

```

case '\b':

    if(TabPtr!=-1&&isTab(p_con->cursor)){//短路原则
        tempBool=1;
    }else if(EnterPtr!=-1&&(tempDelta=isEnter(p_con->cursor))>0) {
        tempBool=2;
    }

    if (p_con->cursor > p_con->original_addr) {
        p_con->cursor--;
        *(p_vmem-2) = ' ';
        *(p_vmem-1) = DEFAULT_CHAR_COLOR;
    }

    if(tempBool==1){//若为TAB 在退掉一格的基础上再退三个格
        for(int j=0;j<3;j++)
            out_char(p_con,'\b');
    }else if(tempBool==2){ //若为TAB 在退掉一格的基础上再退delta-1 格
        for(int j=0;j<tempDelta-1;j++)
            out_char(p_con,'\b');
    }
    break;

```

thank you for reading my introduction.