操作系统实验报告

课程名称：操作系统课程实验

学院名称：计算机科学与技术学院

班　　级：泰山学堂 2016 级

姓　　名：尹小龙

学　　号：201605301355

【实验环境】

1.硬件：　　Lenovo 台式机一机，

2.参考资料：Linux 内核完全注释 0.11，学长学姐的实验报告（无源码）

3.软件环境：Win10 系统，Py/SQL developer，Processing，JavaIDE，Linux_kernel_0.11，虚拟机以及内装的 Ubantu 系统，PowerPoint

【实验内容】

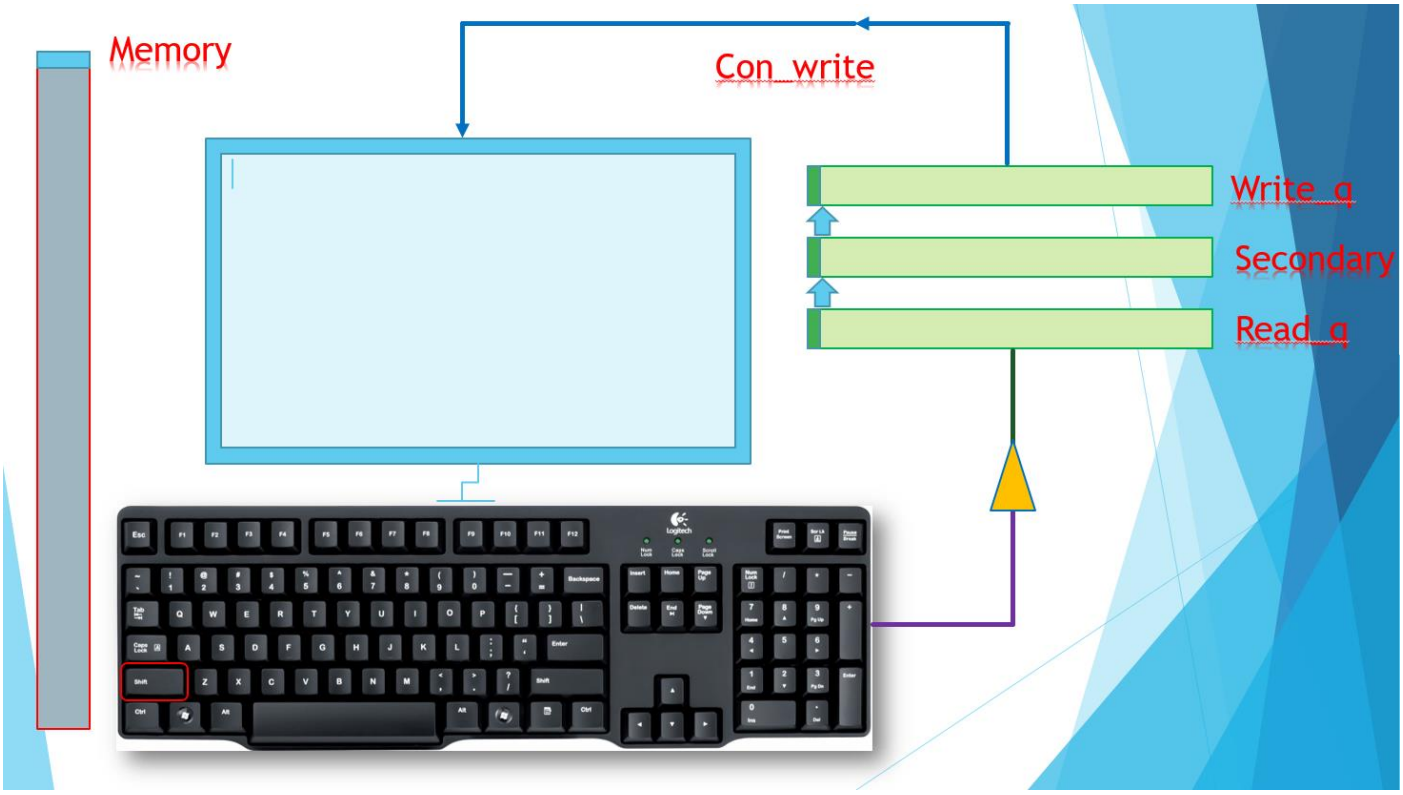
1. 大概方向：在 Linux0.11 中选取一部分或多部分进行解读并将其可视化。
2. 负责部分：Keyboard Interrupt & 屏幕显示
3. 具体介绍：对 Linux0.11 系统中实现键盘中断与屏幕显示部分的代码进行解读，提取数据并将其可视化为容易理解的图形化形式的作品。

【实验过程】

1. 源码解读：负责部分主要涉及源码中的三个文件(tty_io.c\\keyboard.s\\console.c)，其中 tty_io.c 实现了回显和规范模式；keyboard.s 主要包括键盘中断处理程序。在英文惯用法中，make 表示键被按下；break 表示键被松开(放开)。对于 AT 键盘的扫描码，当键按下时，则对应键的扫描码被送出， 但当键松开时，将会发送两个字节，第一个是 0xf0，第 2 个还是按下时的扫描码。为了向下的兼容性，设计人员将 AT 键盘发出的扫描码转换成了老式 PC/XT 标准键盘的扫描码。因此这里仅对 PC/XT 的扫描码进行处理即可；console.c 实现控制台输入输出功能
2. 思路整理：阅读源码之后，发现根据事件间的相互关系，可将模块分为若干事件，每个事件有其独立的逻辑，共划分出来 20 个事件，其中第一个是键盘中断事件。第 2 ~ 19 个是屏幕内存管理事件，合起来是一个大事件，表示屏幕不同变换情况下对应的屏幕开始位置、结束位置、光标位置等数据在屏幕显示内存中的变化情况。最

后一个也是独立事件，表示从缓冲队列中读取键入内容并显示到屏幕上的过程。

（关键帧如下



3. 数据获取：一方面使用了宋振华和陈玉祥组分享的数据提取工具（特此鸣谢！！！），另一方面在阅读源码之后也自己加入了一些中断，具体情况见下表：

1	键盘中断	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"keyboard\",\"data\":{\"chInReadq\":\"%c\",\"current_time\":%d}}\n",c,CURRENT_TIME);
2	跟踪光标当前位置	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"gotoxy\",\"data\":{\"pos\":%d,\"new_x\":%d,\"new_y\":%d}}\n",pos,new_x,new_y);
3	向上卷动一行	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"scrup\",\"data\":{\"scrup_Origin\":%d,\"scrup_Pos\":%d,\"scrup_Scr_end\":%d}}\n",origin,pos,scr_end);
4	向下卷动一行	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"scrdown\",\"data\":{\"scrdown_Origin\":%d,\"scrdown_Pos\":%d,\"scrd

		own_Scr_end\":%d}}\n",origin,pos,scr_end);
5	光标位置 下移一行	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"lf\",\"data\":{\"lf_Origin\":%d,\"lf_Pos\":%d,\"lf_Scr_end\":%d}}\n",origin,pos,scr_end);
6	光标位置 同列下移一行	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"ri\",\"data\":{\"ri_Origin\":%d,\"ri_Pos\":%d,\"ri_Scr_end\":%d}}\n",origin,pos,scr_end);
7	光标回到 第一列	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"cr\",\"data\":{\"cr_Origin\":%d,\"cr_Pos\":%d,\"cr_Scr_end\":%d}}\n",origin,pos,scr_end);
8	擦除光标 前一字符	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"del\",\"data\":{\"del_Origin\":%d,\"del_Pos\":%d,\"del_Scr_end\":%d}}\n",origin,pos,scr_end);
9	删除屏幕上 与光标位置 相关的部分	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"csi_K\",\"data\":{\"csi_K_Origin\":%d,\"csi_K_Pos\":%d,\"csi_K_Scr_end\":%d}}\n",origin,pos,scr_end);
10	删除一行上 与光标位置 相关的部分	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"csi_K\",\"data\":{\"csi_K_Origin\":%d,\"csi_K_Pos\":%d,\"csi_K_Scr_end\":%d}}\n",origin,pos,scr_end);
11	设置显示 字符属性	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"csi_m\",\"data\":{\"csi_m_Origin\":%d,\"csi_m_Pos\":%d,\"csi_m_Scr_end\":%d}}\n",origin,pos,scr_end);
12	在光标处 插入 一空格字符	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"insert_char\",\"data\":{\"insert_ch_Origin\":%d,\"insert_ch_Pos\":%d,\"in

		sert_ch_Scr_end\":%d}}\n",origin,pos,scr_end);
13	在光标处插入一行	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"insert_line\",\"data\":{\"insert_line_Origin\":%d,\"insert_line_Pos\":%d,\"insert_line_Scr_end\":%d}}\n",origin,pos,scr_end);
14	删除一个字符	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"delete_char\",\"data\":{\"deleteCh_Origin\":%d,\"deleteCh_Pos\":%d,\"deleteCh_Scr_end\":%d}}\n",origin,pos,scr_end);
15	删除光标所在行	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"delete_line\",\"data\":{\"delete_line_Origin\":%d,\"delete_line_Pos\":%d,\"delete_line_Scr_end\":%d}}\n",origin,pos,scr_end);
16	在光标出插入 nr 个 字 符	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"csi_at\",\"data\":{\"csi_at_Origin\":%d,\"csi_at_Pos\":%d,\"csi_at_Scr_end\":%d}}\n",origin,pos,scr_end);
17	在光标位置处插入 nr 行	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"csi_L\",\"data\":{\"csi_L_Origin\":%d,\"csi_L_Pos\":%d,\"csi_L_Scr_end\":%d}}\n",origin,pos,scr_end);
18	删除光标位置处的 nr 个字符	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"csi_P\",\"data\":{\"csi_P_Origin\":%d,\"csi_P_Pos\":%d,\"csi_P_Scr_end\":%d}}\n",origin,pos,scr_end);
19	删除光标位置处的 nr 行	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"csi_M\",\"data\":{\"csi_M_Origin\":%d,\"csi_M_Pos\":%d,\"csi_M_Scr_end\":%d}}\n",origin,pos,scr_end);
20	控制台写程序	log("{\"module\":\"interrupt\",\"provider\":\"zl\",\"event\":\"con_write\",\"data\":{\"nr\":%d,\"c\":%c,\"write_Origin\":%d,\"write_Pos

		\":%d,\"write_Scr_end\":%d}}\n",nr,c,origin,pos,scr_end);

数据实例（Json 格式）：


```
{
    "module":"interrupt",
    "provider":"zl",
    "event":"keyboard",
    "data":{
        "chInReadq":"d",
        "current_time":1514015119
    }
}
```


【可视化】

1. 工具选择：因为总是找不到合适的可视化工具，前后试过了 html/flash/PS/ppt，最后考虑到键盘终端部分按键过多的特点选择了 PPT 作为可视化的工具。
2. 设计原理：如下图所示，图片分为显示器，键盘，以及中间的数据传输部分。用户在键盘上按下按键之后，键盘会依据是按键是按下还是弹起向 cpu 发送一个 make 码和一个 break 码，每个按键都有对应的编码（shift 和 alt 键之类的会区分左右，编码不同）。

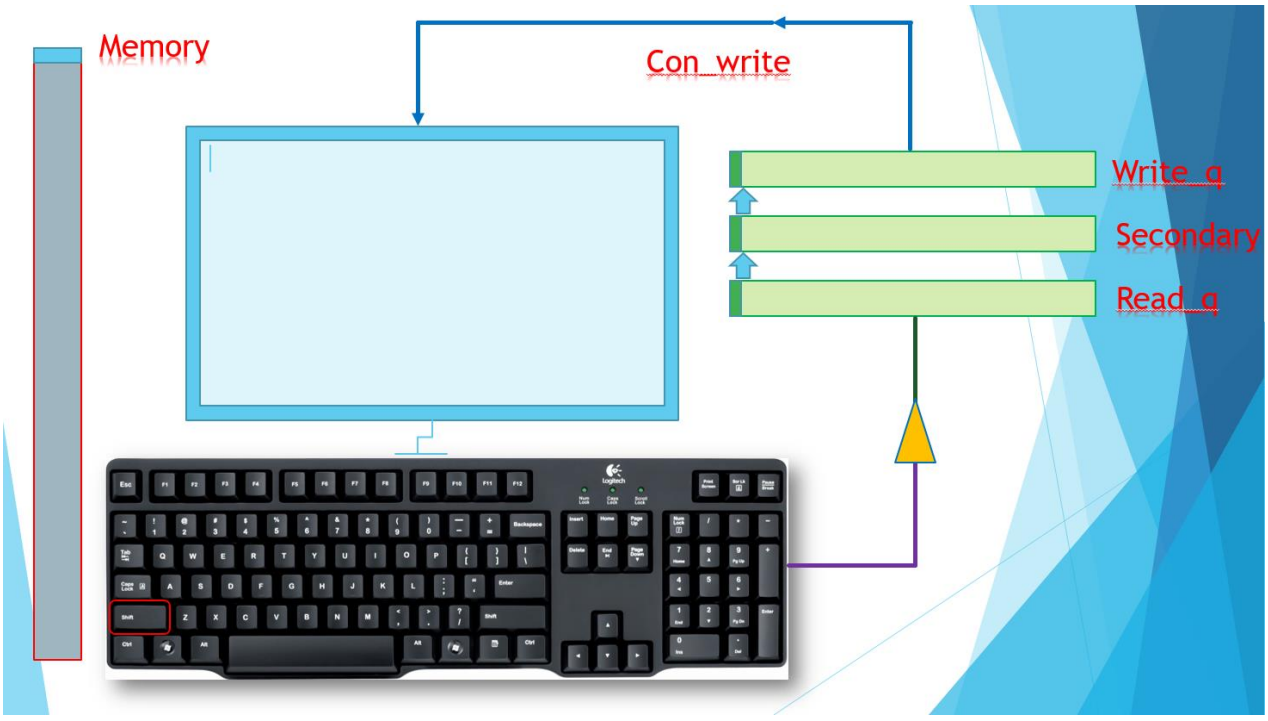
Key	Make	Break	Key	Make	Break
A	1E	9E	N	31	B1
B	30	B0	O	18	98
C	2E	AE	P	19	99
D	20	A0	Q	10	90
E	12	92	R	13	93
F	21	A1	S	1F	9F
G	22	A2	T	14	94
H	23	A3	U	16	96
I	17	97	V	2F	AF
J	24	A4	W	11	91
K	25	A5	X	2D	AD
L	26	A6	Y	15	95
M	32	B2	Z	2C	AC

Key	Make	Break	Key	Make	Break
Backspace	0E	8E	F1	3B	BB
Caps Lock	3A	BA	F2	3C	BC
Enter	1C	9C	F3	3D	BD
Esc	01	81	F4	3E	BE
Left Alt	38	B8	F7	41	C1
Left Ctrl	1D	9D	F5	3F	BF
Left Shift	2A	AA	F6	40	C0
Num Lock	45	C5	F8	42	C2
Right Shift	36	B6	F9	43	C3
Scroll Lock	46	C6	F10	44	C4
Space	39	B9	F11	57	D7
Sys Req (AT)	54	D4	F12	58	D8
Tab	0F	8F			

编码传到 read_q 中，然后经过 Secondary 进行处理变成能被识别的格式

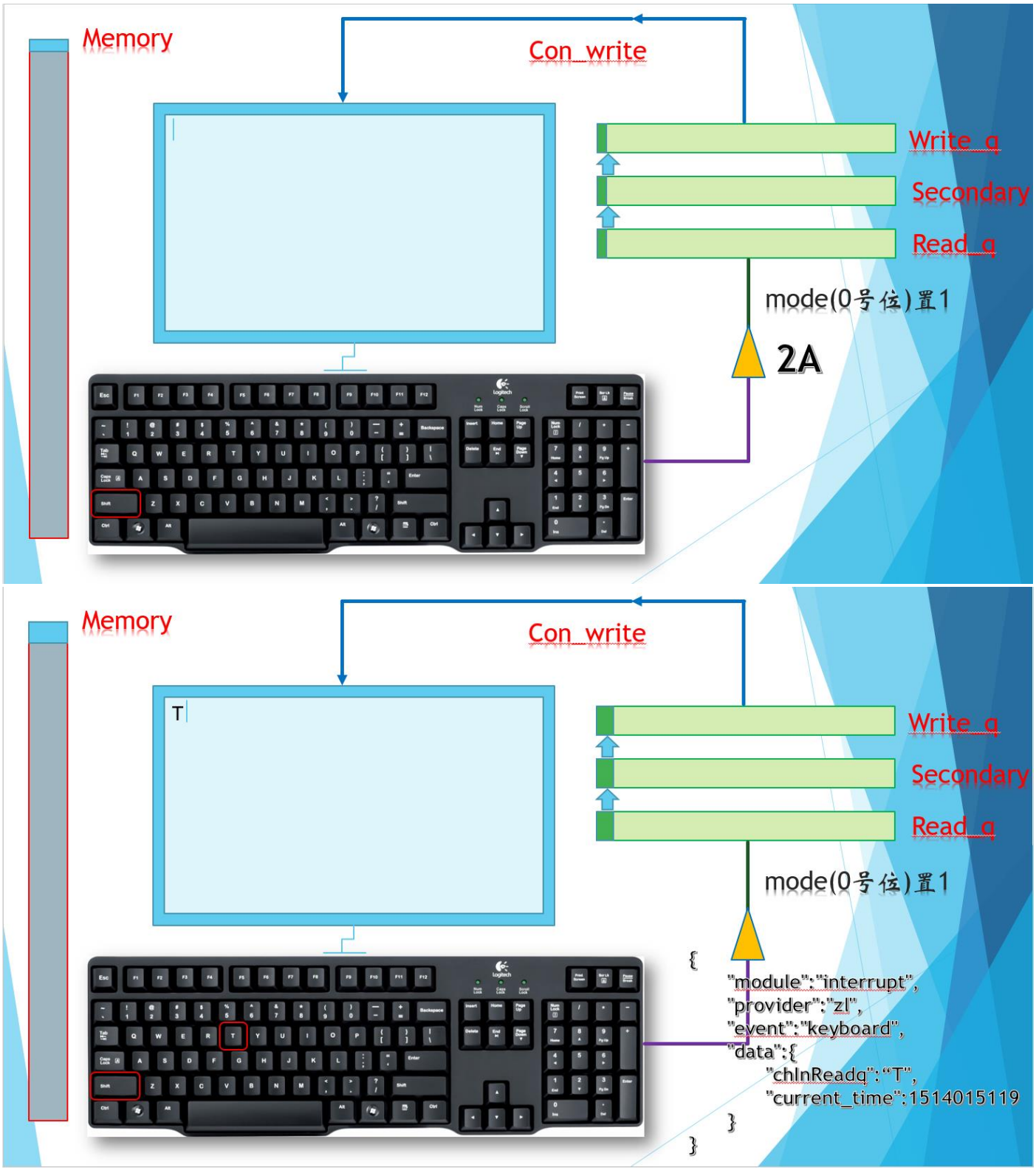

```
{
    "module":"interrupt",
    "provider":"zl",
    "event":"keyboard",
    "data":{
        "chInReadq":"d",
        "current_time":1514015119
    }
}
```

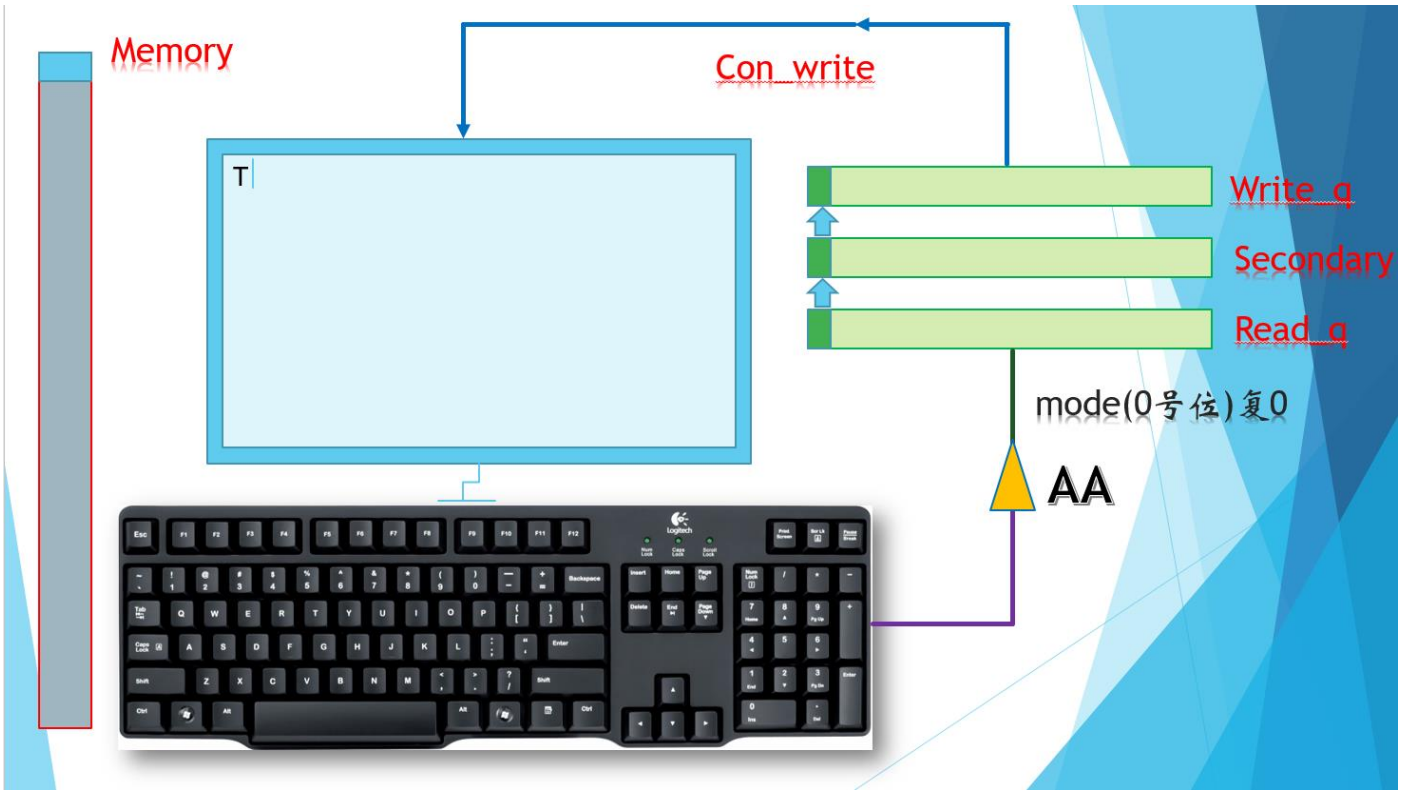

存到 write_q 中，最后通过 Con_write 对屏幕进行输出

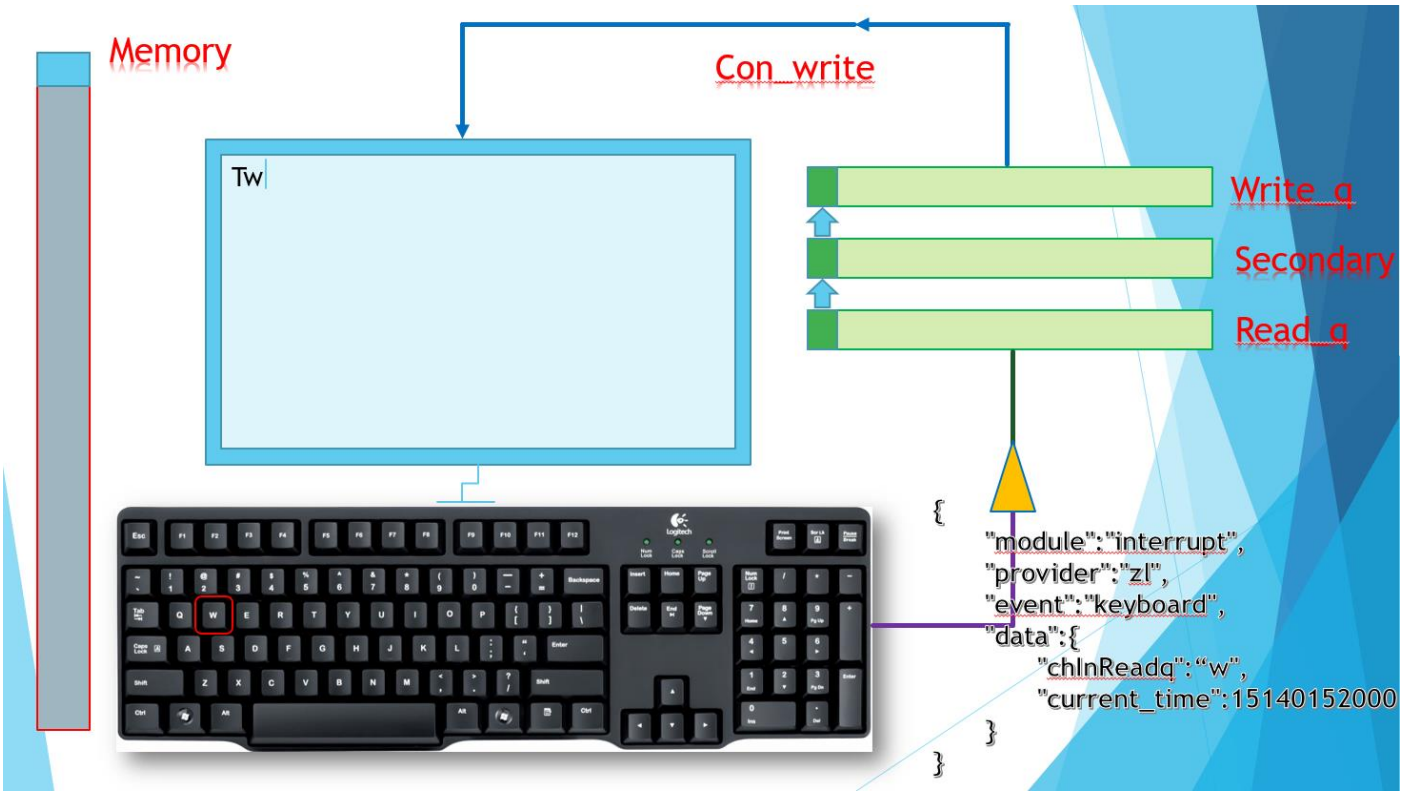


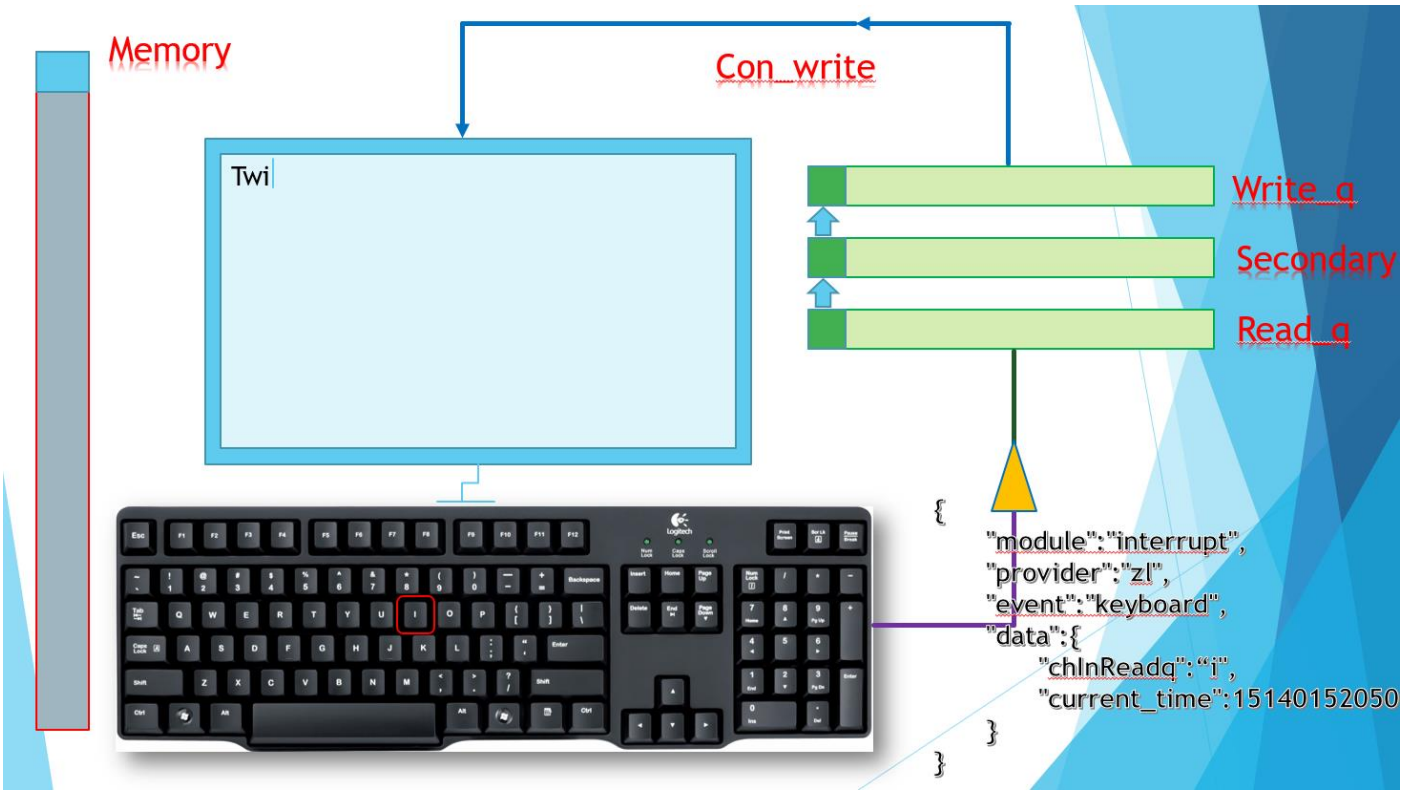
与此同时，对屏幕的输出是由极限的，其大小在左侧的 memory 模块中进行展示。

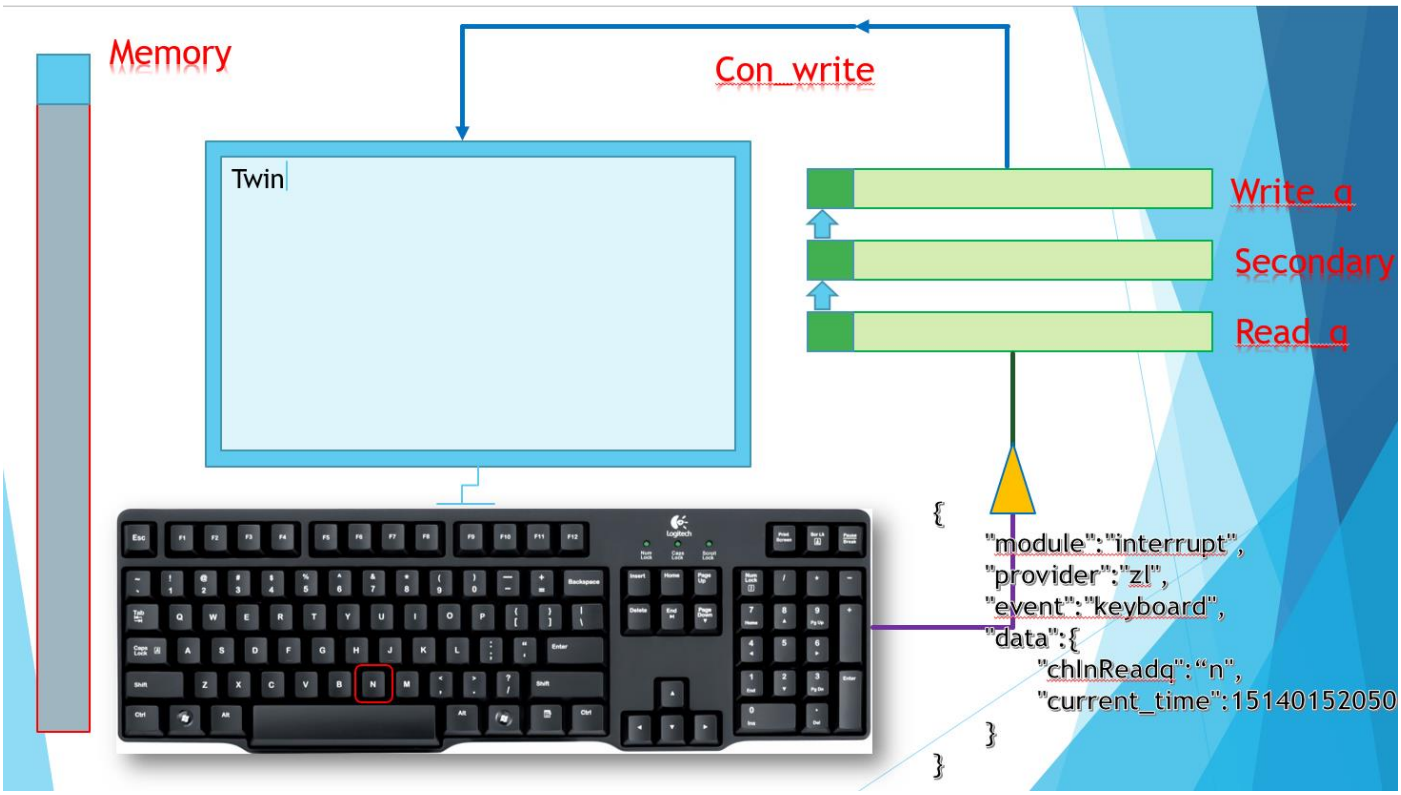
3. 完整截图：

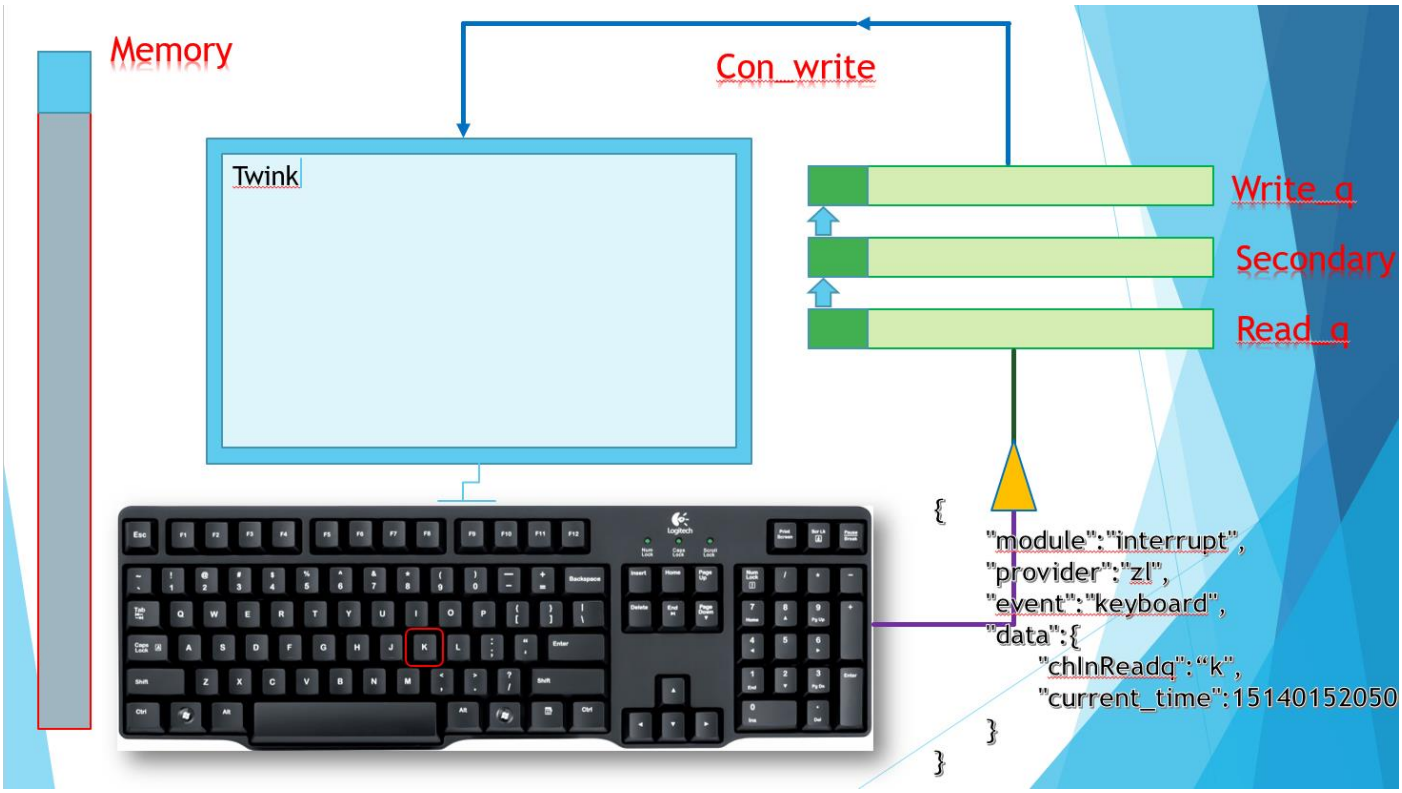


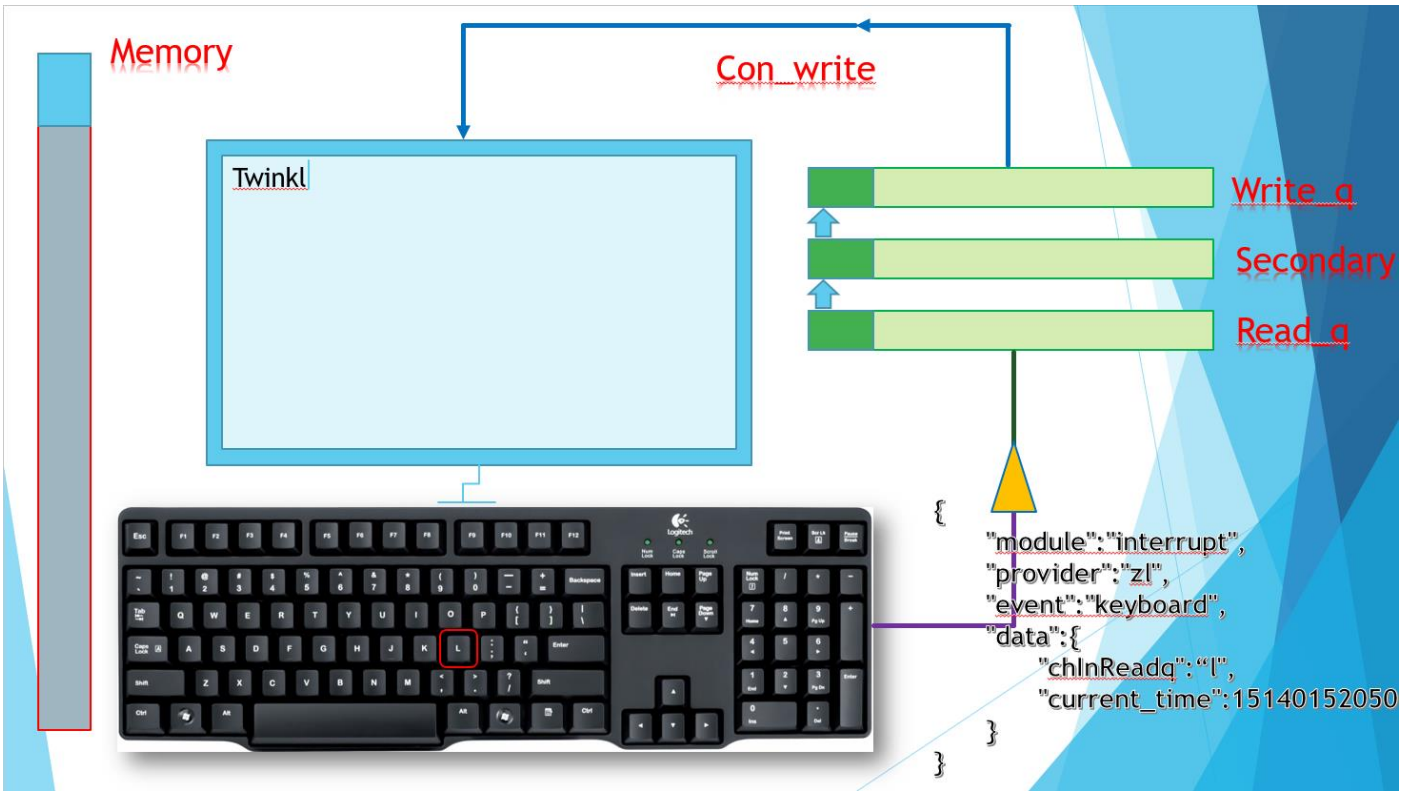

Memory
Con_write
T
Write_q
Secondary
Read_q
mode(0号位)复0
AA

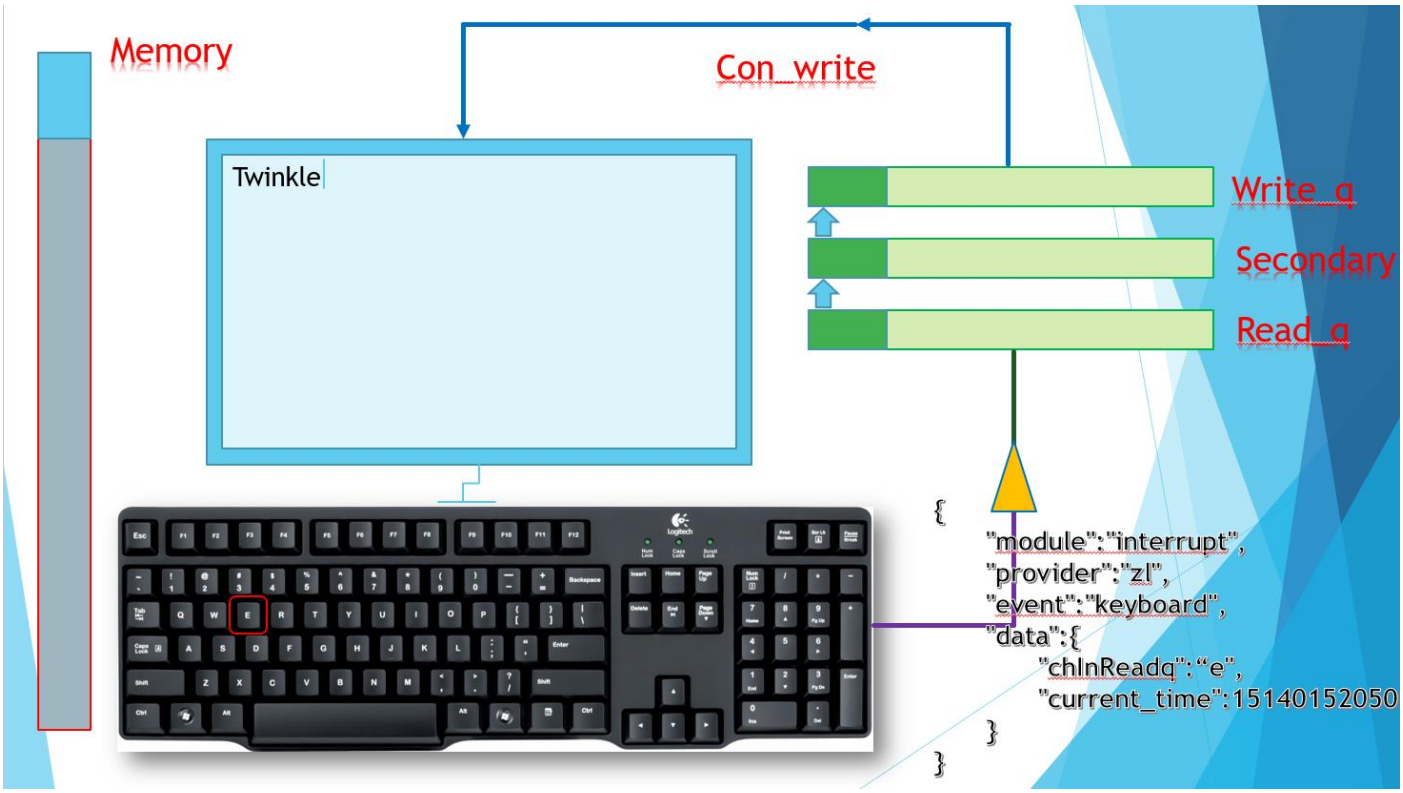

Memory
Con_write
Tw
Write_q
Secondary
Read_q
{
"module":"interrupt",
"provider":"zl",
"event":"keyboard",
"data":{
"chlnReadq":"w",
"current_time":15140152000
}
}

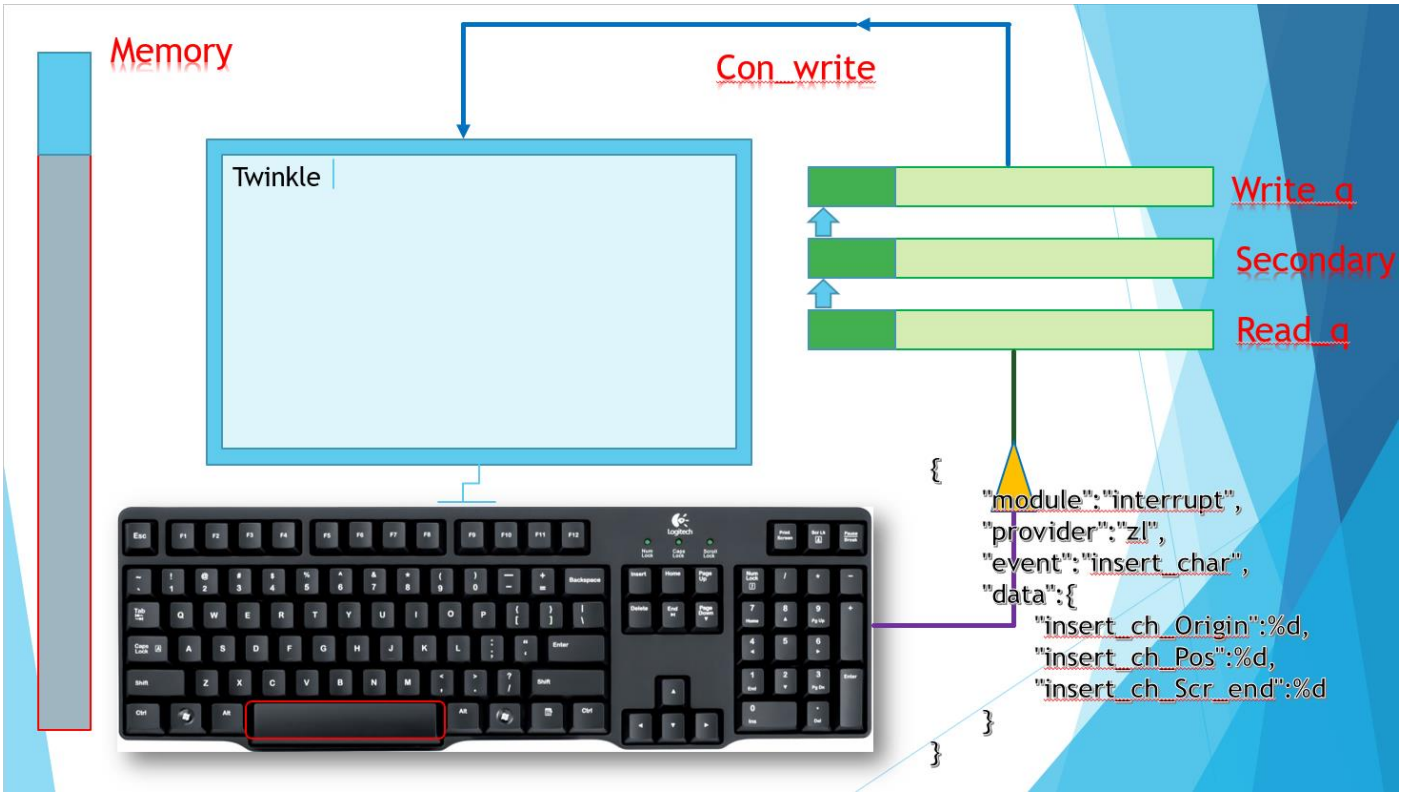

Memory
Con_write
Twi
Write_q
Secondary
Read_q
{
"module":"interrupt",
"provider":"zl",
"event":"keyboard",
"data":{
"chlnReadq":"i",
"current_time":15140152050
}
}

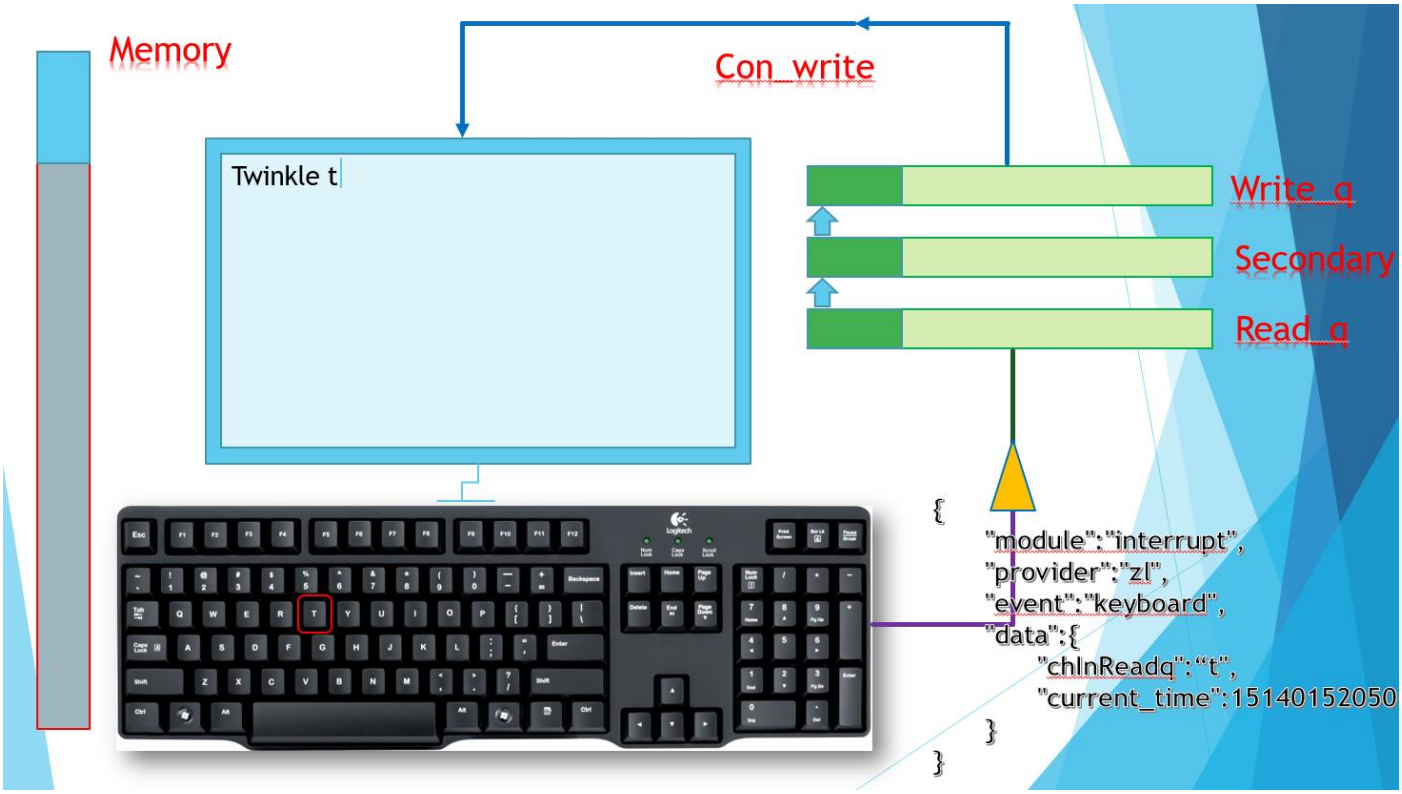

Memory
Con_write
Twin
Write_q
Secondary
Read_q
{
"module":"interrupt",
"provider":"zl",
"event":"keyboard",
"data":{
"chlnReadq":"n",
"current_time":15140152050
}
}

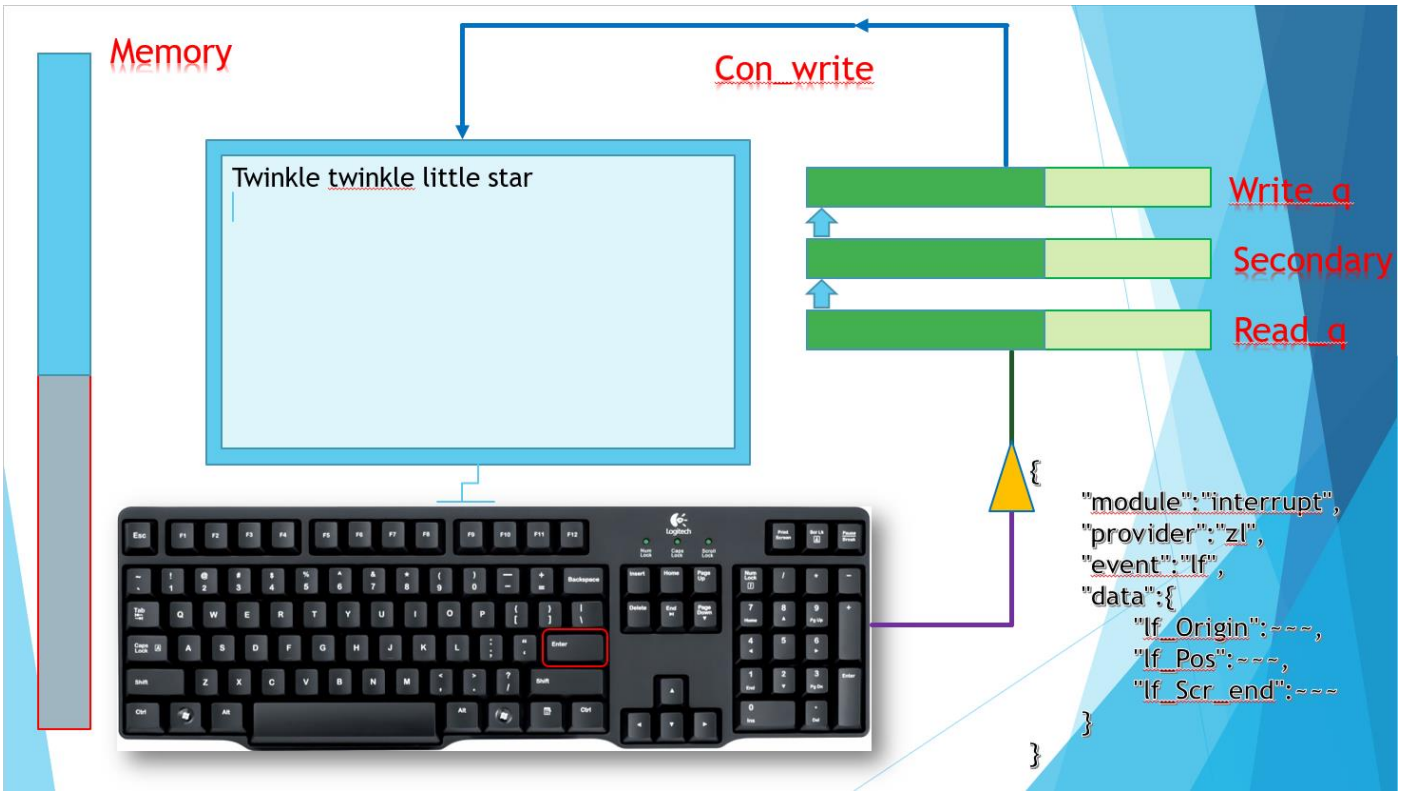

Memory
Con_write
Twink
Write_q
Secondary
Read_q
{
"module":"interrupt",
"provider":"zl",
"event":"keyboard",
"data":{
"chlnReadq":"k",
"current_time":15140152050
}
}

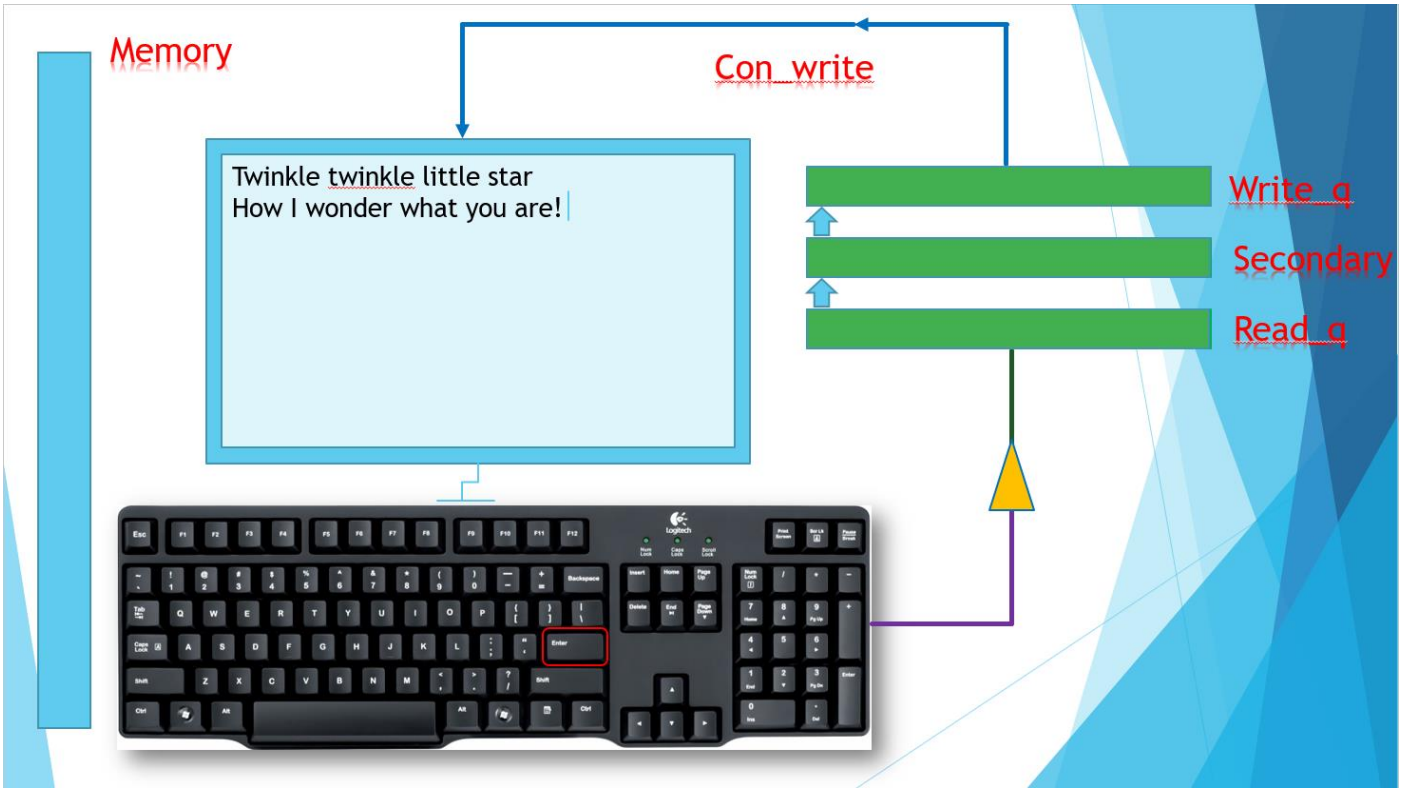

Memory
Con_write
Twinkl
Write_q
Secondary
Read_q
{
"module":"interrupt",
"provider":"zl",
"event":"keyboard",
"data":{
"chlnReadq":"l",
"current_time":15140152050
}
}


Memory
Con_write
Twinkle
Write_q
Secondary
Read_q
{
"module":"interrupt",
"provider":"zl",
"event":"keyboard",
"data":{
"chlnReadq":“e”,
"current_time":15140152050
}
}


Memory
Con_write
Twinkle
Write_q
Secondary
Read_q
{
"module":"interrupt",
"provider":"zl",
"event":"insert_char",
"data":{
"insert_ch_Origin":%d,
"insert_ch_Pos":%d,
"insert_ch_Scr_end":%d
}
}


Memory
Con_write
Twinkle t
Write_q
Secondary
Read_q
{
"module":"interrupt",
"provider":"zl",
"event":"keyboard",
"data":{
"chInReadq":"t",
"current_time":15140152050
}
}


Memory
Con_write
Twinkle twinkle little star
Write_q
Secondary
Read_q
{
"module":"interrupt",
"provider":"zl",
"event":"lf",
"data":{
"lf_Origin":~~~,
"lf_Pos":~~~,
"lf_Scr_end":~~~
}
}



【遇到的问题和解决方案】

数据提取过程中，由于对会变语言等底层语言不了解，而键盘中断部分的文件是 .s 文件，因此曾经一度不知道怎么提取键入的字符，甚至想直接手动构建数据，后来另辟蹊径。在对源码有了一定了解之后，发现可以从缓冲队列中读取数据。 当然，最初由于对数据格式不了解，不知道 JSON 文件如何生成，还有就是对于初接触的 processing 编写代码的时候编码风格不好，导致出现很多冗杂代码等等。

【总结】

通过此次实验，更好的理解了操作系统的运行机理， 虽然只做了一小部分，但是由于需要通读 linux_0.11 源码，所以对其他部分（内存管理，文件系统，块设备驱动等）也有了相关的了解。同时，应可视化软件环境要求，新了解了一门基于 Java 的编程语言 processing，在重新熟悉 Java 的同时， 对图形化的编程产生了兴趣并有了新的进步。虽然编码能力还有欠缺， 在实现可视化的过程中遇到了很多问题，但是最终通过不懈努力完成了动画编程。

【源码粘贴】

```
PImage img, imag, img1, img2, img3, img4, img5, img6;

int num, w = 200, h = 60, s, rows, mo;

PFont myFont;

int board_x, board_y, board_y2;

int k = 0;

int mc_x = width + 1070, mc_y = height / 2 + 600;

String str = "";



class JSONStreamReader {

    private String text;

    private int at;

    private char ch;


    private void next() {

        if (++at < text.length())

            ch = text.charAt(at);

        else

            ch = '\0';

    }


    private void expect(char expected) throws Exception {
```

```
    if (ch == expected) {
        next();
    } else {
        StringBuilder sb = new StringBuilder();
        sb.append("expected character '");
        sb.append(expected);
        sb.append("' instead of ");
        if (at == text.length()) {
            sb.append("EOF");
        } else {
            sb.append('\'');
            sb.append(ch);
            sb.append('\'');
        }
        throw new Exception(sb.toString());
    }
}

private float number() {
    String str = "";

    if (ch == '-') {
        str = "-";
        next();
```

```
}

while (Character.isDigit(ch)) {
    str += ch;
    next();
}

if (ch == '.') {
    str += '.';
    while (Character.isDigit(ch)) {
        str += ch;
    }
}
if (ch == 'e' || ch == 'E') {
    str += ch;
    next();
    if (ch == '-' || ch == '+') {
        str += ch;
        next();
    }
    while (Character.isDigit(ch)) {
        str += ch;
        next();
    }
```

```
    }
    float val = Float.parseFloat(str);
    return val;
}

private void white() {
    while (Character.isWhitespace(ch)) {
        at++;
        if (at == text.length()) {
            break;
        }
        ch = text.charAt(at);
    }
}

private void nil() throws Exception {
    expect('n');
    expect('u');
    expect('l');
    expect('l');
}

private boolean bool() throws Exception {
    if (ch == 't') {
```

```
            expect('t');
            expect('r');
            expect('u');
            expect('e');
            return true;
        } else if (ch == 'f') {
            expect('f');
            expect('a');
            expect('l');
            expect('s');
            expect('e');
            return false;
        } else {
            throw new Exception("bad boolean value");
        }
    }

    private String string() throws Exception {
        String str = "";
        if (ch == "") {
            next();
            while (true) {
                if (ch == "") {
                    next();
```

```
    return str;
}
if (ch == '\\') {
    next();
    if (ch == 'u') {
        throw new Exception("unimplemented escape");
    } else {
        switch (ch) {
        case '"': str += '"'; break;
        case '\\': str += '\\'; break;
        case '/': str += '/'; break;
        case 'b': str += '\b'; break;
        case 'f': str += '\f'; break;
        case 'n': str += '\n'; break;
        case 'r': str += '\r'; break;
        case 't': str += '\t'; break;
        default:
            throw new Exception("unknown escape");
        }
        next();
    }
} else {
    str += ch;
    next();
```

```
            }
        }
    }
    throw new Exception("bad string");
}

private JSONObject object() throws Exception {
    JSONObject obj = new JSONObject();
    if (ch == '{') {
        next();
        white();
        if (ch == '}') {
            next();
            return obj;
        }
        while (true) {
            String key = string();
            white();
            expect(':');
            valueToObject(obj, key);
            white();
            if (ch == '}') {
                next();
                return obj;
```

```
            }
            expect(',');
            white();
        }
    }
    throw new Exception("bad object");
}

private JSONArray array() throws Exception {
    JSONArray result = new JSONArray();
    if (ch == '[') {
        next();
        white();
        if (ch == ']') {
            next();
            return result;
        }
        while (true) {
            valueToArray(result);
            white();
            if (ch == ']') {
                next();
                return result;
            }
```

```
            expect(',');
            white();
        }
    }
    throw new Exception("bad array");
}

private void valueToObject(JSONObject target, String key) throws Exception {
    white();
    switch (ch) {
    case '{':
        target.setJSONObject(key, object());
        break;
    case '[':
        target.setJSONArray(key, array());
        break;
    case '"':
        target.setString(key, string());
        break;
    case '-':
        target.setFloat(key, number());
        break;
    case 't':
    case 'f':
```

```
            target.setBoolean(key, bool());
            break;
        case 'n':
            nil();
            target.setJSONObject(key, null);
            break;
        default:
            if (Character.isDigit(ch))
                target.setFloat(key, number());
            else
                throw new Exception("expected a value after the key");
        }
    }

    private void valueToArray(JSONArray target) throws Exception {
        white();
        switch (ch) {
        case '{':
            target.append(object());
            break;
        case '[':
            target.append(array());
            break;
        case '"':
```

```
        target.append(string());
        break;
    case '-':
        target.append(number());
        break;
    case 't':
    case 'f':
        target.append(bool());
        break;
    case 'n':
        nil();
        target.append(0);
        target.setJSONObject(target.size() - 1, null);
        break;
    default:
        if (Character.isDigit(ch))
            target.append(number());
        else
            throw new Exception("expected a valid value in array");
    }
}

public JSONStreamReader(String filename) {
    byte[] bytes = loadBytes(filename);
```

```
        text = new String(bytes);
        at = 0;
        ch = text.charAt(0);
    }

    public JSONObject nextObject() throws Exception {
        white();
        return object();
    }

    public boolean hasMore() {
        white();
        return at < text.length();
    }

}

void test() {
    JSONStreamReader s = new JSONStreamReader("test.json");
    try {
        while (s.hasMore()) {
            JSONObject obj = s.nextObject();
            print("\n\n= = = = = = = = = =\n\n");
            print(obj);
```

```
        }

    } catch (Exception e) {
        print(e);
        e.printStackTrace();
    }
}

void setup(){

    // *scale
    size(1600, 1000);
    //frameRate(2);



    //the location to display the scan code
    board_x = 1100;
    board_y = 300;
    board_y2 = 320;
    s = 0;
}

void draw(){
    background(0);
```

```
    //#####################################################
    //keyboard data from os_kbjson
    //#####################################################

    JSONStreamReader jsr = new JSONStreamReader("E:\\work_file\\os\\os_kb_2.json");
    //JSONStreamReader jsr = new JSONStreamReader("C:\\Users\\ 朱 丽 \\Desktop\\data.json");
    try {
        while (jsr.hasMore()) {
            JSONObject obj = jsr.nextObject();
            print(obj);
            String ch = obj.getString("chInReadq");
            str += ch;
            //String event = obj.getString("event");
            //print(event);
        }
    } catch (Exception e) {
        print(e);
        e.printStackTrace();
    }

    //frameRate(2);
    img4 = loadImage("E:/work_file/os/cousor.jpg");
```

```
img5 = loadImage("E:/work_file/os/xianshi1.jpg");
img6 = loadImage("E:/work_file/os/jiantou.jpg");//the cousor

pushMatrix();
translate(width/2, height/2 - 150);
num = 0;

//the memory, read_q, secondary and write_q
for(int i = 0; i < 10; i++){
    stroke(0, 200, 0);
    rect(-600, num-200, w, h); //memory
    num += h;
}
rect(300, -300, width/3, 40);
rect(300, -220, width/3, 40);
rect(300, -140, width/3, 40);
textSize(30);
//textAlign(CENTER);
text("Memery", -650, -250);
text("write_q", 600, -300);
text("secondary", 600, -220);
text("read_q", 600, -140);
```

```
//progress of the memory and queues
noStroke();
fill(100, 100, 200, 150);
rect(-600, -230+s/2, w, s);//memory
if(s<width/3)
{
   rect(30+s/2, -300, s, 40);//read_q
   rect(30+s/2, -220, s, 40);//secondary
   rect(30+s/2, -140, s, 40);//wrote_q
}

//little display screen
image(img5, -470, -230 + s);

//the change of origin, pos and scr_end in memory
s = s+1;
if(s < 100)
{
   stroke(150);
   line(-700, -230, -500, -230);
   image(img4, -470, -230);
}
if(s > 100 && -200 + s / 2 < num)
{
```

```
        stroke(150);
        line(-700, -300 + s, -500, -300 + s);
        image(img4, -470, -300 + s);
    }
    if(s>200 && -200+s/2 < num)
    {
        stroke(255);
        line(-700, -400+s, -500, -400+s);
        image(img5, -470, -420+s);
        //text(, -400, -400+s);      //data from JSON
    }
    if(s > num){
        s = 0;
        stroke(255);
        line(-700, -400+s, -500, -400+s);
    }

    strokeWeight(3);
    stroke(255, 0, 0);
    //triangle(25+s%(width/3),-170,30+s%(width/3),-190,35+s%(width/3),-170);  //the  trangle
between read_q and second_q
    image(img6, 10+s%(width/3), -200);
    image(img6, 10+s%(width/3), -280);
    //triangle(25+s%(width/3),-260,30+s%(width/3),-280,35+s%(width/3),-260);
```

```
stroke(255);
line(0, height/4-20, 0, height/4+20);
line(100, height/4+20, -100, height/4+20);
stroke(150);
line(200, height/3, 500, height/3);
line(500, height/3, 500, -height/9);
//triangle(500,height/6,490,height/6+30,510,height/6+30);//arrows on the line
line(-100, -height/3+30, 30, -height/3+30);
line(-100, -height/3+30, -100, -height/9);
//triangle(-100,-height/6,-110,-height/6-30,-90,-height/6-30); //arrows on the line

//computer console
stroke(255);
noFill();
rect(0, height/16, width/3, height/3);
rect(0, height/16, width/3-30, height/3-30);

String s1 = "he x2 and y2 define a rectand y2 parameters";
String s2 = "This fun to work with";
String s3 = "parameters define a rectangular arand";
String s4 = "y2 parameters define a rect";
String s5 = "angular arparameters";
```

```
String s6 = "define a rectangular    y2";

String s7 = "parameters and y2 define a rectangular ar ";

String s8 = "parameters define a rectangular";

String s9 = "ar and y define";

String s10 = "a rectangular display define a rect";

String s11 = "withifont will area to be used instead";


//text on the console

fill(random(100), random(250), random(200));

myFont = createFont("Georgia", 25);

textFont(myFont);

translate(-width/2, -height/2+150);


rows = (s * 3) % 260;

if(rows < 1)

{

    //text(s1,560,270-rows);

    text(s2, 560, 295 - rows);

    text(s3, 560, 320 - rows);

    text(s4, 560, 345 - rows);

    text(s5, 560, 370 - rows);

    text(s6, 560, 395 - rows);

    text(s7, 560, 420 - rows);

    text(s8, 560, 445 - rows);
```

```
    text(s9, 560, 470 - rows);
    text(s10, 560, 495 - rows);
    text(s11, 560, 520 - rows);
}
else if(rows>1 && rows<25)
{
    //text(s2,560,295-rows);
    text(s3, 560, 320 - rows);
    text(s4, 560, 345 - rows);
    text(s5, 560, 370 - rows);
    text(s6, 560, 395 - rows);
    text(s7, 560, 420 - rows);
    text(s8, 560, 445 - rows);
    text(s9, 560, 470 - rows);
    text(s10, 560, 495 - rows);
    text(s11, 560, 520 - rows);
    text(s1, 560, 545 - rows);
}
else if(rows>25 && rows<50)
{
    //text(s3,560,320-rows);
    text(s4, 560, 345 - rows);
    text(s5, 560, 370 - rows);
    text(s6, 560, 395 - rows);
```

```
    text(s7, 560, 420 - rows);
    text(s8, 560, 445 - rows);
    text(s9, 560, 470 - rows);
    text(s10, 560, 495 - rows);
    text(s11, 560, 520 - rows);
    text(s1, 560, 545 - rows);
    text(s2, 560, 570 - rows);
}
    else if(rows>50 && rows<75)
{
    //text(s4,560,345-rows);
    text(s5, 560, 370 - rows);
    text(s6, 560, 395 - rows);
    text(s7, 560, 420 - rows);
    text(s8, 560, 445 - rows);
    text(s9, 560, 470 - rows);
    text(s10, 560, 495 - rows);
    text(s11, 560, 520 - rows);
    text(s1, 560, 545 - rows);
    text(s2, 560, 570 - rows);
    text(s3, 560, 595 - rows);
}
    else if(rows>75 && rows<100)
{
```

```
    //text(s5,560,370-rows);
    text(s6, 560, 395 - rows);
    text(s7, 560, 420 - rows);
    text(s8, 560, 445 - rows);
    text(s9, 560, 470 - rows);
    text(s10, 560, 495 - rows);
    text(s11, 560, 520 - rows);
    text(s1, 560, 545 - rows);
    text(s2, 560, 570 - rows);
    text(s3, 560, 595 - rows);
    text(s4, 560, 620 - rows);
}

    else if(rows>100 && rows<125)
{
    //text(s6,560,395-rows);
    text(s7, 560, 420 - rows);
    text(s8, 560, 445 - rows);
    text(s9, 560, 470 - rows);
    text(s10, 560, 495 - rows);
    text(s11, 560, 520 - rows);
    text(s1, 560, 545 - rows);
    text(s2, 560, 570 - rows);
    text(s3, 560, 595 - rows);
```

```
    text(s4, 560, 620 - rows);
    text(s5, 560, 645 - rows);
}

    else if(rows>125 && rows<150)
{
    //text(s7,560,420-rows);
    text(s8, 560, 445 - rows);
    text(s9, 560, 470 - rows);
    text(s10, 560, 495 - rows);
    text(s11, 560, 520 - rows);
    text(s1, 560, 545 - rows);
    text(s2, 560, 570 - rows);
    text(s3, 560, 595 - rows);
    text(s4, 560, 620 - rows);
    text(s5, 560, 645 - rows);
    text(s6, 560, 670 - rows);
}

    else if(rows>150&&rows<175)
{
    //text(s7,560,420-rows);
    text(s8, 560, 445 - rows);
    text(s9, 560, 470 - rows);
```

```
    text(s10, 560, 495 - rows);
    text(s11, 560, 520 - rows);
    text(s1, 560, 545 - rows);
    text(s2, 560, 570 - rows);
    text(s3, 560, 595 - rows);
    text(s4, 560, 620 - rows);
    text(s5, 560, 645 - rows);
    text(s6, 560, 670 - rows);
}

    else if(rows>175&&rows<200)
{
    //text(s8,560,445-rows);
    text(s9, 560, 470 - rows);
    text(s10, 560, 495 - rows);
    text(s11, 560, 520 - rows);
    text(s1, 560, 545 - rows);
    text(s2, 560, 570 - rows);
    text(s3, 560, 595 - rows);
    text(s4, 560, 620 - rows);
    text(s5, 560, 645 - rows);
    text(s6, 560, 670 - rows);
    text(s7, 560, 695 - rows);
}
```

```
    else if(rows>200 && rows<225)
{
    //text(s9,560,470-rows);
    text(s10, 560, 495 - rows);
    text(s11, 560, 520 - rows);
    text(s1, 560, 545 - rows);
    text(s2, 560, 570 - rows);
    text(s3, 560, 595 - rows);
    text(s4, 560, 620 - rows);
    text(s5, 560, 645 - rows);
    text(s6, 560, 670 - rows);
    text(s7, 560, 695 - rows);
    text(s8, 560, 720 - rows);
}
    else if(rows>225 && rows<250)
{
    //text(s10,560,495-rows);
    text(s11, 560, 520 - rows);
    text(s1, 560, 545 - rows);
    text(s2, 560, 570 - rows);
    text(s3, 560, 595 - rows);
    text(s4, 560, 620 - rows);
    text(s5, 560, 645 - rows);
    text(s6, 560, 670 - rows);
```

```
        text(s7, 560, 695 - rows);
        text(s8, 560, 720 - rows);
        text(s9, 560, 745 - rows);
    }
    translate(width/2, height/2 - 150);
    fill(100, 100, 200);

    //imag=loadImage("C:\\Users\\朱丽\\Downloads\\sound.gif");
    //image(imag,-200,-60);

    rectMode(CENTER);
    noFill();
    stroke(255);
    popMatrix();

    translate(width/4, height/2 + height/8);
    img=loadImage("E:/work_file/os/keyboard.jpg");
    img1=loadImage("E:/work_file/os/loudou6.jpg");      //direction of char,from keyboard to
read_q
    img2=loadImage("E:/work_file/os/loudou7.jpg");      //from write_q to console
    img3=loadImage("E:/work_file/os/finger.jpg");      //pressed char on keyboard
    image(img1, 860, -200);
    image(img2, 275, -500);
    //image(img4, 300, -500);
```

```
image(img, 0, 0);



translate(-width/6, -height/2 - height/8);
//textSize(25);
//textAlign(CENTER);
//text("Enjoy your hacking with\nLinux!", width/3, height/8);



int len = str.length();
print("str=" + str + "\n");
//for(int i=0;i<len;i++)
//{
//    print("str(i)"+str.charAt(i)+"\n");
//}
//print("len:",len);

if(k < len)
{
    print("str(k)=" + str.charAt(k) + '\n');
    //frameRate(30);
    mo = mo + 5;
    if(mo > 400)
```

```
{
   mo = 0;
}
//println('(',mouseX,',',mouseY,')');
noStroke();
fill(300, 100, 100);

if(str.charAt(k) == '\n')
{
   image(img3, 720, 805);    //a little hand on the keyboard
   text("enter\n", board_x, board_y2);
   text("", mc_x, mc_y-mo);
}
//-----------------------------------------------------Q
if(str.charAt(k) == 'q'||str.charAt(k) == 'Q')
{
   //textAlign(CENTER);
   image(img3, 330, 780);    //a little hand on the keyboard
   text("Q: single--10      shift--10\n", board_x, board_y);
   text("      ctrl--10      alt--10\n", board_x, board_y2);
   text("Q", mc_x, mc_y-mo);
}

//-----------------------------------------------------W
```

```
  if(str.charAt(k) == 'w'||str.charAt(k) == 'W')
{
    //textAlign(CENTER);
    image(img3, 360, 780);
    text("W: single--11      shift--11\n", board_x, board_y);
    text("      ctrl--11      alt--11\n", board_x, board_y2);
    text("W", mc_x, mc_y-mo);
}

//-----------------------------------------------------E
  if(str.charAt(k) == 'e'||str.charAt(k) == 'E')
{
    //textAlign(CENTER);
    image(img3, 390, 780);
    text("E: single--12      shift--12\n", board_x, board_y);
    text("      ctrl--12      alt--12\n", board_x, board_y2);
    text("E", mc_x, mc_y-mo);
}

//-----------------------------------------------------R
  if(str.charAt(k) == 'r'||str.charAt(k) == 'R')
{
    //textAlign(CENTER);
    image(img3, 425, 780);
```

```
    text("R: single--13      shift--13\n", board_x, board_y);
    text("      ctrl--13      alt--13\n", board_x, board_y2);
    text("R", mc_x, mc_y-mo);
}

//-------------------------------------------------T
  if(str.charAt(k) == 't'||str.charAt(k) == 'T')
{
    //textAlign(CENTER);
    image(img3, 460, 780);
    text("T: single--14      shift--14\n", board_x, board_y);
    text("      ctrl--14      alt--14\n", board_x, board_y2);
    text("T", mc_x, mc_y-mo);
}

//-------------------------------------------------Y
  if(str.charAt(k) == 'y'||str.charAt(k) == 'Y')
{
    //textAlign(CENTER);
    image(img3, 490, 780);
    text("Y: single--15      shift--15\n", board_x, board_y);
    text("      ctrl--15      alt--15\n", board_x, board_y2);
    text("Y", mc_x, mc_y-mo);
}
```

```
//-------------------------------------------------U
  if(str.charAt(k) == 'u'||str.charAt(k) == 'U')
{
    //textAlign(CENTER);
    image(img3, 520, 780);
    text("U: single--16      shift--16\n", board_x, board_y);
    text("      ctrl--16      alt--16\n", board_x, board_y2);
    text("U", mc_x, mc_y-mo);
}

//-------------------------------------------------I
  if(str.charAt(k)=='i'||str.charAt(k)=='I')
{
    //textAlign(CENTER);
    image(img3, 555, 780);
    text("I: single--17      shift--17\n", board_x, board_y);
    text("      ctrl--17      alt--17\n", board_x, board_y2);
    text("I", mc_x, mc_y-mo);
}

//-------------------------------------------------O
  if(str.charAt(k) == 'o'||str.charAt(k) == 'O')
{
```

```
    //textAlign(CENTER);
    image(img3, 585, 780);
    text("O: single--18      shift--18\n", board_x, board_y);
    text("      ctrl--18      alt--18\n", board_x, board_y2);
    text("O", mc_x, mc_y-mo);
}

//-------------------------------------------------P
  if(str.charAt(k) == 'p'||str.charAt(k) == 'P')
{
    //textAlign(CENTER);
    image(img3, 620, 780);
    text("P: single--19      shift--19\n", board_x, board_y);
    text("      ctrl--19      alt--19\n", board_x, board_y2);
    text("P", mc_x, mc_y-mo);
}

  //-----------------------------------------------------A
if(str.charAt(k) == 'a'||str.charAt(k) == 'A')
{
    //textAlign(CENTER);
    image(img3, 335, 815);
    text("A: single--1E      shift--1E\n",board_x,board_y);
    text("      ctrl--1E      alt--1E\n",board_x,board_y2);
```

```
    text("A",mc_x,mc_y-mo);
}
  //-----------------------------------------------------S
if(str.charAt(k)=='s'||str.charAt(k)=='S')
{
    //textAlign(CENTER);
    image(img3,370,815);
    text("S: single--1F      shift--1F\n",board_x,board_y);
    text("      ctrl--1F      alt--1F\n",board_x,board_y2);
    text("S",mc_x,mc_y-mo);
}
  //-----------------------------------------------------D
if(str.charAt(k)=='d'||str.charAt(k)=='D')
{
    //textAlign(CENTER);
    image(img3,400,815);
    text("D: single--20      shift--20\n",board_x,board_y);
    text("      ctrl--20      alt--20\n",board_x,board_y2);
    text("D",mc_x,mc_y-mo);
}
  //-----------------------------------------------------F
if(str.charAt(k)=='f'||str.charAt(k)=='F')
{
    //textAlign(CENTER);
```

```
    image(img3,430,815);
    text("F: single--21      shift--21\n",board_x,board_y);
    text("      ctrl--21      alt--21\n",board_x,board_y2);
    text("F",mc_x,mc_y-mo);
}
  //-------------------------------------------------------G
if(str.charAt(k)=='g'||str.charAt(k)=='G')
{
    //textAlign(CENTER);
    image(img3,460,815);
    text("G: single--22      shift--22\n",board_x,board_y);
    text("      ctrl--22      alt--22\n",board_x,board_y2);
    text("G",mc_x,mc_y-mo);
}
  //-------------------------------------------------------H
if(str.charAt(k)=='h'||str.charAt(k)=='H')
{
    //textAlign(CENTER);
    image(img3,490,815);
    text("H: single--23      shift--23\n",board_x,board_y);
    text("      ctrl--23      alt--23\n",board_x,board_y2);
    text("H",mc_x,mc_y-mo);
}
  //-------------------------------------------------------J
```

```
if(str.charAt(k)=='j'||str.charAt(k)=='J')
{
   //textAlign(CENTER);
   image(img3,520,815);
   text("J: single--24      shift--24\n",board_x,board_y);
   text("      ctrl--24      alt--24\n",board_x,board_y2);
   text("J",mc_x,mc_y-mo);
}
  //-------------------------------------------------------K
if(str.charAt(k)=='k'||str.charAt(k)=='K')
{
   //textAlign(CENTER);
   image(img3,555,815);
   text("K: single--25      shift--25\n",board_x,board_y);
   text("      ctrl--25      alt--25\n",board_x,board_y2);
   text("K",mc_x,mc_y-mo);
}
  //-------------------------------------------------------L
if(str.charAt(k)=='l'||str.charAt(k)=='L')
{
   //textAlign(CENTER);
   image(img3,590,815);
   text("L: single--26      shift--26\n",board_x,board_y);
   text("      ctrl--26      alt--26\n",board_x,board_y2);
```

```
    text("L",mc_x,mc_y-mo);
}
  //-----------------------------------------------------Z
if(str.charAt(k)=='z'||str.charAt(k)=='Z')
{
    //textAlign(CENTER);
    image(img3,360,845);
    text("Z: single--2C      shift--2C\n",board_x,board_y);
    text("      ctrl--2C      alt--2C\n",board_x,board_y2);
    text("Z",mc_x,mc_y-mo);
}
  //-----------------------------------------------------X
if(str.charAt(k)=='x'||str.charAt(k)=='X')
{
    //textAlign(CENTER);
    image(img3,390,845);
    text("X: single--2D      shift--2D\n",board_x,board_y);
    text("      ctrl--2D      alt--2D\n",board_x,board_y2);
    text("X",mc_x,mc_y-mo);
}
  //-----------------------------------------------------C
if(str.charAt(k)=='c'||str.charAt(k)=='C')
{
    //textAlign(CENTER);
```

```
    image(img3,425,845);
    text("C: single--2E      shift--2E\n",board_x,board_y);
    text("      ctrl--2E      alt--2E\n",board_x,board_y2);
    text("C",mc_x,mc_y-mo);
}
  //-------------------------------------------------V
if(str.charAt(k)=='v'||str.charAt(k)=='V')
{
    //textAlign(CENTER);
    image(img3,455,845);
    text("V: single--2F      shift--2F\n",board_x,board_y);
    text("      ctrl--2F      alt--2F\n",board_x,board_y2);
    text("V",mc_x,mc_y-mo);
}
  //-------------------------------------------------B
if(str.charAt(k)=='b'||str.charAt(k)=='B')
{
    //textAlign(CENTER);
    image(img3,490,845);
    text("B: single--30      shift--30\n",board_x,board_y);
    text("      ctrl--30      alt--30\n",board_x,board_y2);
    text("B",mc_x,mc_y-mo);
}
  //-------------------------------------------------N
```

```
if(str.charAt(k)=='n'||str.charAt(k)=='N')
{
   //textAlign(CENTER);
   image(img3,520,845);
   text("N: single--31      shift--31\n",board_x,board_y);
   text("      ctrl--31      alt--31\n",board_x,board_y2);
   text("N",mc_x,mc_y-mo);
}
  //-----------------------------------------------------M
if(str.charAt(k)=='m'||str.charAt(k)=='M')
{
   //textAlign(CENTER);
   image(img3,555,845);
   text("M: single--32      shift--32\n",board_x,board_y);
   text("      ctrl--32      alt--32\n",board_x,board_y2);
   text("M",mc_x,mc_y-mo);
}
  //-----------------------------------------------------' '
if(str.charAt(k)==' ')
{
   //textAlign(CENTER);
   text("Blank",board_x,board_y);
}
k=k+1;
```

```
    noFill();
    stroke(255);
}
k=k+1;

//#############################################################
//actions when the keyboard is pressed
//mo:speed of moving
//img3:a little hand on the keyboard
//#############################################################
mo=mo+5;
if(mo>400)
{
    mo=0;
}
if(mousePressed)
{
    //println('(',mouseX,',',mouseY,')');
    noStroke();
    fill(300,100,100);

    //a little hand on the keyboard
    image(img3,mouseX-150,mouseY);
```

```
//-----------------------------------------------------Q
if(mouseX>465&&mouseX<495&&mouseY>765&&mouseY<795)
{
   //textAlign(CENTER);
   text("Q: single--10      shift--10\n",board_x,board_y);
   text("      ctrl--10      alt--10\n",board_x,board_y2);
   text("Q",mc_x,mc_y-mo);
}

//-----------------------------------------------------W
  if(mouseX>500&&mouseX<525&&mouseY>765&&mouseY<795)
{
   //textAlign(CENTER);
   text("W: single--11      shift--11\n",board_x,board_y);
   text("      ctrl--11      alt--11\n",board_x,board_y2);
   text("W",mc_x,mc_y-mo);
}

//-----------------------------------------------------E
  if(mouseX>530&&mouseX<555&&mouseY>765&&mouseY<795)
{
   //textAlign(CENTER);
   text("E: single--12      shift--12\n",board_x,board_y);
   text("      ctrl--12      alt--12\n",board_x,board_y2);
```

```
    text("E",mc_x,mc_y-mo);
}

//-------------------------------------------------R
  if(mouseX>560&&mouseX<590&&mouseY>765&&mouseY<795)
{
    //textAlign(CENTER);
    text("R: single--13      shift--13\n",board_x,board_y);
    text("      ctrl--13      alt--13\n",board_x,board_y2);
    text("R",mc_x,mc_y-mo);
}

//-------------------------------------------------T
  if(mouseX>595&&mouseX<620&&mouseY>765&&mouseY<795)
{
    //textAlign(CENTER);
    text("T: single--14      shift--14\n",board_x,board_y);
    text("      ctrl--14      alt--14\n",board_x,board_y2);
    text("T",mc_x,mc_y-mo);
}

//-------------------------------------------------Y
  if(mouseX>625&&mouseX<650&&mouseY>765&&mouseY<795)
{
```

```
    //textAlign(CENTER);
    text("Y: single--15      shift--15\n",board_x,board_y);
    text("      ctrl--15      alt--15\n",board_x,board_y2);
    text("Y",mc_x,mc_y-mo);
}

//-------------------------------------------------U
  if(mouseX>655&&mouseX<685&&mouseY>765&&mouseY<795)
{
    //textAlign(CENTER);
    text("U: single--16      shift--16\n",board_x,board_y);
    text("      ctrl--16      alt--16\n",board_x,board_y2);
    text("U",mc_x,mc_y-mo);
}

//-----------------------------------------------I
  if(mouseX>695&&mouseX<715&&mouseY>765&&mouseY<795)
{
    //textAlign(CENTER);
    text("I: single--17      shift--17\n",board_x,board_y);
    text("      ctrl--17      alt--17\n",board_x,board_y2);
    text("I",mc_x,mc_y-mo);
}
```

```
//-----------------------------------------------O
  if(mouseX>720&&mouseX<750&&mouseY>765&&mouseY<795)
{
    //textAlign(CENTER);
    text("O: single--18      shift--18\n",board_x,board_y);
    text("      ctrl--18      alt--18\n",board_x,board_y2);
    text("O",mc_x,mc_y-mo);
}

//-----------------------------------------------P
  if(mouseX>755&&mouseX<780&&mouseY>765&&mouseY<795)
{
    //textAlign(CENTER);
    text("P: single--19      shift--19\n",board_x,board_y);
    text("      ctrl--19      alt--19\n",board_x,board_y2);
    text("P",mc_x,mc_y-mo);
}

  //-----------------------------------------------------A
if(mouseX>475&&mouseX<495&&mouseY>800&&mouseY<830)
{
    //textAlign(CENTER);
    text("A: single--1E      shift--1E\n",board_x,board_y);
    text("      ctrl--1E      alt--1E\n",board_x,board_y2);
```

```
    text("A",mc_x,mc_y-mo);
}
  //-----------------------------------------------------S
if(mouseX>505&&mouseX<535&&mouseY>800&&mouseY<830)
{
    //textAlign(CENTER);
    text("S: single--1F      shift--1F\n",board_x,board_y);
    text("      ctrl--1F      alt--1F\n",board_x,board_y2);
    text("S",mc_x,mc_y-mo);
}
  //-----------------------------------------------------D
if(mouseX>535&&mouseX<560&&mouseY>800&&mouseY<830)
{
    //textAlign(CENTER);
    text("D: single--20      shift--20\n",board_x,board_y);
    text("      ctrl--20      alt--20\n",board_x,board_y2);
    text("D",mc_x,mc_y-mo);
}
  //-----------------------------------------------------F
if(mouseX>565&&mouseX<590&&mouseY>800&&mouseY<830)
{
    //textAlign(CENTER);
    text("F: single--21      shift--21\n",board_x,board_y);
    text("      ctrl--21      alt--21\n",board_x,board_y2);
```

```
    text("F",mc_x,mc_y-mo);
}
  //-------------------------------------------------------G
if(mouseX>595&&mouseX<625&&mouseY>800&&mouseY<830)
{
    //textAlign(CENTER);
    text("G: single--22      shift--22\n",board_x,board_y);
    text("      ctrl--22      alt--22\n",board_x,board_y2);
    text("G",mc_x,mc_y-mo);
}
  //-------------------------------------------------------H
if(mouseX>630&&mouseX<655&&mouseY>800&&mouseY<830)
{
    //textAlign(CENTER);
    text("H: single--23      shift--23\n",board_x,board_y);
    text("      ctrl--23      alt--23\n",board_x,board_y2);
    text("H",mc_x,mc_y-mo);
}
  //-------------------------------------------------------J
if(mouseX>660&&mouseX<690&&mouseY>800&&mouseY<830)
{
    //textAlign(CENTER);
    text("J: single--24      shift--24\n",board_x,board_y);
    text("      ctrl--24      alt--24\n",board_x,board_y2);
```

```
    text("J",mc_x,mc_y-mo);
}
  //-------------------------------------------------------K
if(mouseX>695&&mouseX<715&&mouseY>800&&mouseY<830)
{
    //textAlign(CENTER);
    text("K: single--25      shift--25\n",board_x,board_y);
    text("      ctrl--25      alt--25\n",board_x,board_y2);
    text("K",mc_x,mc_y-mo);
}
  //-------------------------------------------------------L
if(mouseX>725&&mouseX<755&&mouseY>800&&mouseY<830)
{
    //textAlign(CENTER);
    text("L: single--26      shift--26\n",board_x,board_y);
    text("      ctrl--26      alt--26\n",board_x,board_y2);
    text("L",mc_x,mc_y-mo);
}
  //-------------------------------------------------------Z
if(mouseX>500&&mouseX<525&&mouseY>830&&mouseY<860)
{
    //textAlign(CENTER);
    text("Z: single--2C      shift--2C\n",board_x,board_y);
    text("      ctrl--2C      alt--2C\n",board_x,board_y2);
```

```
    text("Z",mc_x,mc_y-mo);
}
  //-------------------------------------------------------X
if(mouseX>530&&mouseX<555&&mouseY>830&&mouseY<860)
{
    //textAlign(CENTER);
    text("X: single--2D      shift--2D\n",board_x,board_y);
    text("      ctrl--2D      alt--2D\n",board_x,board_y2);
    text("X",mc_x,mc_y-mo);
}
  //-------------------------------------------------------C
if(mouseX>560&&mouseX<590&&mouseY>830&&mouseY<860)
{
    //textAlign(CENTER);
    text("C: single--2E      shift--2E\n",board_x,board_y);
    text("      ctrl--2E      alt--2E\n",board_x,board_y2);
    text("C",mc_x,mc_y-mo);
}
  //-------------------------------------------------------V
if(mouseX>595&&mouseX<620&&mouseY>830&&mouseY<860)
{
    //textAlign(CENTER);
    text("V: single--2F      shift--2F\n",board_x,board_y);
    text("      ctrl--2F      alt--2F\n",board_x,board_y2);
```

```
    text("V",mc_x,mc_y-mo);
}
  //-------------------------------------------------------B
if(mouseX>625&&mouseX<650&&mouseY>830&&mouseY<860)
{
    //textAlign(CENTER);
    text("B: single--30      shift--30\n",board_x,board_y);
    text("      ctrl--30      alt--30\n",board_x,board_y2);
    text("B",mc_x,mc_y-mo);
}
  //-------------------------------------------------------N
if(mouseX>655 && mouseX<680 && mouseY>830 && mouseY<860)
{
    //textAlign(CENTER);
    text("N: single--31      shift--31\n",board_x,board_y);
    text("      ctrl--31      alt--31\n",board_x,board_y2);
    text("N",mc_x,mc_y-mo);
}
  //-------------------------------------------------------M
if(mouseX>690 && mouseX<720 && mouseY>830 && mouseY<860)
{
    //textAlign(CENTER);
    text("M: single--32      shift--32\n",board_x,board_y);
    text("      ctrl--32      alt--32\n",board_x,board_y2);
```

```
        text("M",mc_x,mc_y-mo);
      }
       //-----------------------------------------------------''
      if(mouseX>550 && mouseX<730 && mouseY>855 && mouseY<910)
      {
        //textAlign(CENTER);
        text("Blank",board_x,board_y);
      }
      noFill();
      stroke(255);
    }
}
```