

# Reporte de practica 9

González Pardo Adrian

Abril 2020

## 1. Código fuente:

```
1  /*
2   * Alumno: Gonzalez Pardo Adrian
3   * Grupo: 3CV8
4   * Practica 9
5   */
6  #include <bits/stdc++.h>
7
8  using namespace std;
9  class Stack{
10 private:
11     bool UP,DW,WPC,CLR;
12     unsigned short PC[16],SP;
13     unsigned short PCin,PCout;
14 public:
15     Stack(){
16         srand(time(NULL));
17     }
18     bool getUP(){
19         return this->UP;
20     }
21
22     bool getDW(){
23         return this->DW;
24     }
25
26     bool getWPC(){
27         return this->WPC;
28     }
29
30     bool getCLR(){
31         return this->CLR;
32     }
33
34     unsigned short *getPC(){
35         return this->PC;
36     }
37
38     unsigned short getSP(){
39         return this->SP;
40     }
41
42     unsigned short getPCin(){
43         return this->PCin;
44     }
45
46     unsigned short getPCout(){
47         return this->PCout;
48     }
49
50     void setUP(bool UP){
51         this->UP=UP;
```

```

52 }
53
54 void setDW(bool DW){
55     this->DW=DW;
56 }
57
58 void setWPC(bool WPC){
59     this->WPC=WPC;
60 }
61
62 void setCLR(bool CLR){
63     this->CLR=CLR;
64 }
65
66 void setSP(unsigned short SP){
67     this->SP=SP;
68 }
69
70 void setPCin(unsigned short PCin){
71     this->PCin=PCin;
72 }
73
74 void setPCout(unsigned short PCout){
75     this->PCout=PCout;
76 }
77
78 void set(){
79     for(int i=0;i<16;i++){
80         *(PC+i)=(unsigned short)(rand()%65535)-16384;
81     }
82     setSP(0);
83 }
84 void get(){
85     for(int i=0;i<16;i++){
86         printf("\tPC[%d] = %d\n",i,*(PC+i));
87     }
88 }
89
90 void operacion(unsigned short PCin,bool UP,bool DW,bool WPC, bool CLR){
91     setPCin(PCin);
92     setUP(UP);
93     setDW(DW);
94     setWPC(WPC);
95     setCLR(CLR);
96     if(getCLR()){
97         setSP(0);
98         for(int i=0;i<16;i++){
99             *(PC+i)=0;
100         }
101         PCout=*(PC+getSP());
102         return;
103     }
104     if(!getWPC()&&!getUP()&&!getDW()){
105         setSP(getSP());
106         *(PC+getSP())+=1;
107         PCout=*(PC+getSP());
108         return ;
109     }
110     if(getWPC()&&!getUP()&&!getDW()){
111         setSP(getSP());
112         *(PC+getSP())=PCin;
113         PCout=*(PC+getSP());
114         return;
115     }
116     if(getWPC()&&getUP()&&!getDW()){
117         if(getSP()+1<16){
118             setSP(getSP()+1);

```

```

119     }
120     *(PC+getSP())=PCin;
121     PCout=*(PC+getSP());
122     return;
123 }
124 if(!getWPC() && !getUP() && getDW()){
125     if(getSP() > 0){
126         setSP(getSP() - 1);
127     }
128     *(PC+getSP()) += 1;
129     PCout = *(PC+getSP());
130     return;
131 }
132
133 }
134
135 void operacion(){
136     printf("PC[%d] = %d\n", getSP(), *(PC+getSP()));
137 }
138 };
139 int main(int argc, char *argv[]) {
140     Stack st;
141     printf("Set()\n");
142     st.set();
143     st.get();
144     printf("CLR\n");
145     st.operacion(0,0,0,0,1);
146     st.get();
147     st.operacion();
148     printf("Instruccion\n");
149     printf("LI R6, #87 \t");
150     st.operacion(0,0,0,0,0);
151     st.operacion();
152     printf("LI R8, #90 \t");
153     st.operacion(0,0,0,0,0);
154     st.operacion();
155     printf("B 34 \t");
156     st.operacion(34,0,0,1,0);
157     st.operacion();
158     printf("ADD R8, R2, R3 \t");
159     st.operacion(0,0,0,0,0);
160     st.operacion();
161     printf("SUB R1, R2, R3 \t");
162     st.operacion(0,0,0,0,0);
163     st.operacion();
164     printf("CALL 0x61 \t");
165     st.operacion(61,1,0,1,0);
166     st.operacion();
167     printf("LI R6, #87 \t");
168     st.operacion(0,0,0,0,0);
169     st.operacion();
170     printf("LI R8, #90 \t");
171     st.operacion(0,0,0,0,0);
172     st.operacion();
173     printf("CALL 100 \t");
174     st.operacion(100,1,0,1,0);
175     st.operacion();
176     printf("ADD R8, R2, R3 \t");
177     st.operacion(0,0,0,0,0);
178     st.operacion();
179     printf("SUB R1, R2, R3 \t");
180     st.operacion(0,0,0,0,0);
181     st.operacion();
182     printf("LI R6, #87 \t");
183     st.operacion(0,0,0,0,0);
184     st.operacion();
185     printf("RET \t");

```

```

186 st.operacion(0,0,1,0,0);
187 st.operacion();
188 printf("SUB R1, R2, R3 \t");
189 st.operacion(0,0,0,0,0);
190 st.operacion();
191 printf("LI R6, #87 \t");
192 st.operacion(0,0,0,0,0);
193 st.operacion();
194 printf("RET \t");
195 st.operacion(0,0,1,0,0);
196 st.operacion();
197 printf("B 300 \t");
198 st.operacion(300,0,0,1,0);
199 st.operacion();
200 printf("CALL 889 \t");
201 st.operacion(889,1,0,1,0);
202 st.operacion();
203 printf("ADD R8, R2, R3 \t");
204 st.operacion(0,0,0,0,0);
205 st.operacion();
206 printf("SUB R1, R2, R3 \t");
207 st.operacion(0,0,0,0,0);
208 st.operacion();
209 printf("LI R6, #87 \t");
210 st.operacion(0,0,0,0,0);
211 st.operacion();
212 printf("RET \t");
213 st.operacion(0,0,1,0,0);
214 st.operacion();
215 printf("RET \t");
216 st.operacion(0,0,1,0,0);
217 st.operacion();
218 printf("Get\n");
219 st.get();
220 return 0;
221 }

```

## 2. Simulación (Screenshots)

```
Set()
PC[0] = 54060
PC[1] = 9343
PC[2] = 52267
PC[3] = 45774
PC[4] = 16923
PC[5] = 57067
PC[6] = 49718
PC[7] = 58081
PC[8] = 48595
PC[9] = 57100
PC[10] = 1326
PC[11] = 35913
PC[12] = 26187
PC[13] = 18197
PC[14] = 46660
PC[15] = 41446

CLR
PC[0] = 0
PC[1] = 0
PC[2] = 0
PC[3] = 0
PC[4] = 0
PC[5] = 0
PC[6] = 0
PC[7] = 0
PC[8] = 0
PC[9] = 0
PC[10] = 0
PC[11] = 0
PC[12] = 0
PC[13] = 0
PC[14] = 0
PC[15] = 0
```

Figura 0: con instrucción en la captura de pantalla

```
PC[0] = 0
Instruccion
LI R6, #87 PC[0] = 1
```

Figura 1: con instrucción en la captura de pantalla

```
LI R8, #90 PC[0] = 2
```

Figura 2: con instrucción en la captura de pantalla

```
B 34 PC[0] = 34
```

Figura 3: con instrucción en la captura de pantalla

```
ADD R8, R2, R3 PC[0] = 35
```

Figura 4: con instrucción en la captura de pantalla

```
SUB R1, R2, R3 PC[0] = 36
```

Figura 5: con instrucción en la captura de pantalla

```
CALL 0x61 PC[1] = 61
```

Figura 6: con instrucción en la captura de pantalla

```
LI R6, #87 PC[1] = 62
```

Figura 7: con instrucción en la captura de pantalla

```
LI R8, #90 PC[1] = 63
```

Figura 8: con instrucción en la captura de pantalla

```
CALL 100 PC[2] = 100
```

Figura 9: con instrucción en la captura de pantalla

```
ADD R8, R2, R3 PC[2] = 101
```

Figura 10: con instrucción en la captura de pantalla

SUB R1, R2, R3 PC[2] = 102

Figura 11: con instrucción en la captura de pantalla

LI R6, #87 PC[2] = 103

Figura 12: con instrucción en la captura de pantalla

RET PC[1] = 64

Figura 13: con instrucción en la captura de pantalla

SUB R1, R2, R3 PC[1] = 65

Figura 14: con instrucción en la captura de pantalla

LI R6, #87 PC[1] = 66

Figura 15: con instrucción en la captura de pantalla

RET PC[0] = 37

Figura 16: con instrucción en la captura de pantalla

B 300 PC[0] = 300

Figura 17: con instrucción en la captura de pantalla

CALL 889 PC[1] = 889

Figura 18: con instrucción en la captura de pantalla

ADD R8, R2, R3 PC[1] = 890

Figura 19: con instrucción en la captura de pantalla

SUB R1, R2, R3 PC[1] = 891

Figura 20: con instrucción en la captura de pantalla

LI R6, #87 PC[1] = 892

Figura 21: con instrucción en la captura de pantalla

RET PC[0] = 301

Figura 22: con instrucción en la captura de pantalla

RET PC[0] = 302

Figura 23: con instrucción en la captura de pantalla

```
Get
PC[0] = 302
PC[1] = 892
PC[2] = 103
PC[3] = 0
PC[4] = 0
PC[5] = 0
PC[6] = 0
PC[7] = 0
PC[8] = 0
PC[9] = 0
PC[10] = 0
PC[11] = 0
PC[12] = 0
PC[13] = 0
PC[14] = 0
PC[15] = 0
```

Figura 24: con instrucción en la captura de pantalla