

Hongwen Fan

cs211

pa4

This assignment is absolutely the most funny and challenging assignment that I have ever met. The ideology of this program is not complicated but the implementation task is extreme heavy cause there are bunch of y86 instruction there and we have to make a code for every instruction .

My frame is very simple . I just make a CPU structure to store all the registers and all the flags. The initial part of the program just follow professor's way in the class .And when I execute the instruction in store in the main memory , I use a instruction counter which is part of the CPU structure to track my running of the y86. The running time is just linear .And the space cost is exactly the size of the y86 program.

Then I finish all the extra credit included disassembler assembler and make my own y86 program which is to compute the Fibonacci number . It is a definitely stuff.

By the way , if you wanna extra my output of disassembler and assembler to a txt file , you just use the > symbol to the terminal just like the picture shown .You will create a txt file to store the assembly instruction output .Then you are good to go with the assembler .

```
[hf164@cs211-il final]$ ./y86dis prog2.y86 >assembly.txt
```

And this is the content of the assembly.txt. The left most column is the location of the instruction in the memory.

100	irmovl	1800	%esp
106	rrmovl	%esp	%ebp
108	call	14a	
10d	halt		
10e	nop		
10f	pushl	%ebp	
111	rrmovl	%esp	%ebp
113	irmovl	10	%eax
119	subl	%eax	%esp
11b	rmmovl	%edi	fffffffc(%ebp)
121	irmovl	1	%eax
127	andl	%edi	%edi
129	je	145	
12e	rrmovl	fffffffc(%ebp)	%ebx
134	subl	%eax	%ebx
136	rrmovl	%ebx	%edi
138	call	10f	
13d	rrmovl	fffffffc(%ebp)	%ebx
143	mull	%ebx	%eax
145	rrmovl	%ebp	%esp
147	popl	%ebp	
149	ret		
14a	pushl	%ebp	
14c	rrmovl	%esp	%ebp
14e	irmovl	4	%ebx

The usage for the assembler is that you have to use the txt file created by the y86dis as a

input file to the y86ass, and the y86ass will give you a bunch of .text number which is a part of y86 file . I just use stdout to show the output .You must get my point, right ? I hope you can enjoy my program . <3

```
[hf164@c211-il final]$ ./y86dis prog2.y86 >assembly.txt
[hf164@c211-il final]$ ./y86ass assembly.txt
.text    100      30f4001800002045804a0100001000a05f204530f01000000061044075fcffff
ff30f0010000000627773450100005035fcffffff61032037800f0100005035fcffffff64302054b0
5f90a05f204530f30400000030f00100000030f212000000d03f0000000060036102746001000030
f120000000c11f0000000073ae010000507100000000800f01000030f124000000400100000000d1
1f0000000030f128000000d01f00000000704e0100002054b05f90
[hf164@c211-il final]$
```