VE370
HW3
Bingcheng
516021910219

2.21.4:
adjust location should be addi $sp, $sp, -8 #空间
Negative
and free the stack should be addi $sp, $sp, 8
positive.

2.21.5:

$s0 = a2
v0 = s0 + f(a2)+

```
int f(a, b, c) {
    return g(a,b)+c
}
```

2.21.6
500 + C = 500 + 1000 = 1500

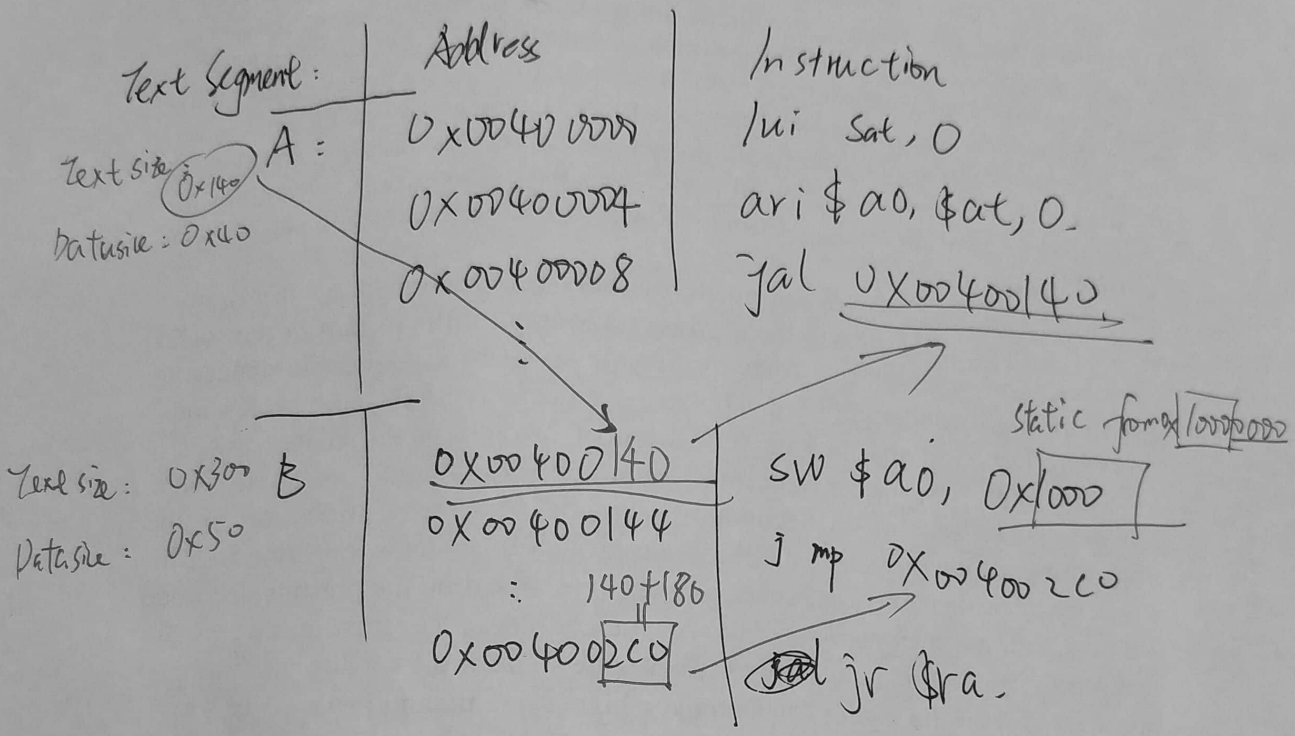
2.31.1

	text size	data size
A	0x140	0x40
B	0x300	0x50

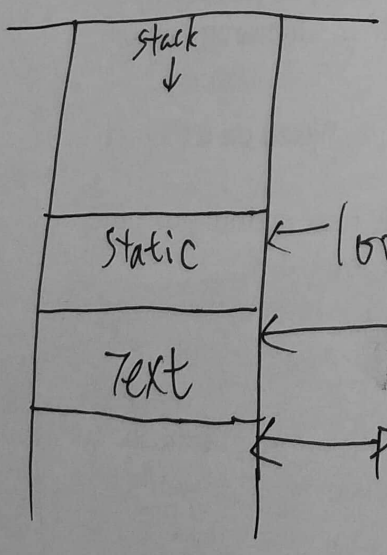
Executable File: A+B:

text size = 0x140 + 0x300 = 0x440

Data size = 0x40 + 0x50 = 0x90.



2.31.2.



Data: 1000 8000 - 1000 0000

= 8000 hex

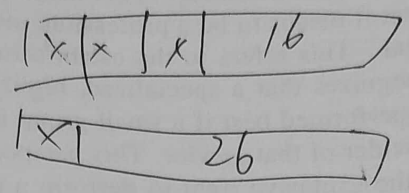
Tex: 1000 0000 - 0040 0000

= 0xFFC0 0000

2.31.3:

If objects are linked too far apart,
It's impossible to use branch & jump instr.

Branch : 16 bits,
Jump : 26 bits.



7.

```
Module mux(A,B,sel,F); input
[31:0] A;
Input [31:0] B;
input sel;
output [31:0] F; reg[31:0] F;
always @(sel or A or B) begin
if (sel == 0) F = A;
else F = B; end
endmodule
```

8.

```
module register(rdreg1,rdreg2,wrreg,wrdata,write,rddata1,rddata2); input [4:0]rdreg1;
input [4:0]rdreg2;
input [4:0]wrreg;
input [31:0]wrdata;
input write;
output[31:0]rddata1;
output[31:0]rddata2;
reg [31:0]data[0:31];
reg [31:0]rddata1;
reg [31:0]rddata2;
always @(posedge write) begin
data[wrreg]=wrdata; end
always @(rdreg1) begin
assign rddata1=data[rdreg1]; end
always @(rdreg2) begin
assign rddata2=data[rdreg2]; end
endmodule
```