

Jan 25, 11 15:29

I03-handout.txt

Page 1/2

```

1 Handout for CS 372H
2 Lecture 3
3 25 January 2011
4
5 gcc calling conventions and PC emulation
6
7 [Credit to Frans Kaashoek, Robert Morris, and Nickolai Zeldovich.]
8
9 1. Example for gcc calling conventions
10
11 Here is some C code:
12 int main(void) { return f(8)+1; }
13 int f(int x) { return g(x); }
14 int g(int x) { return x+3; }
15
16 Here is assembly code that might correspond to this C code:
17
18 _main:
19     prologue
20
21     pushl %ebp
22     movl %esp, %ebp
23
24     body
25     pushl $8
26     call _f
27     addl $1, %eax
28
29     epilogue
30     movl %ebp, %esp
31     popl %ebp
32     ret
33
34 _f:
35     prologue
36
37     pushl %ebp
38     movl %esp, %ebp
39
40     body
41     pushl 8(%esp)
42     call _g
43
44     epilogue
45     movl %ebp, %esp
46     popl %ebp
47     ret
48
49 <small version of _g>:
50     movl 4(%esp), %eax
51     addl $3, %eax
52     ret
53
54 <longer version of _g>:
55     prologue
56     pushl %ebp
57     movl %esp, %ebp
58
59     save %ebx
60     pushl %ebx
61
62     body
63     movl 8(%ebp), %ebx
64     addl $3, %ebx
65     movl %ebx, %eax
66
67     restore %ebx
68     popl %ebx
69
70     epilogue
71     movl %ebp, %esp
72     popl %ebp
73     ret

```

Jan 25, 11 15:29

I03-handout.txt

Page 2/2

```

74
75 2. Emulation of CPU in software
76
77 for (;;) {
78     read_instruction();
79     switch (decode_instruction_opcode()) {
80     case OPCODE_ADD:
81         int src = decode_src_reg();
82         int dst = decode_dst_reg();
83         regs[dst] = regs[dst] + regs[src];
84         break;
85     case OPCODE_SUB:
86         int src = decode_src_reg();
87         int dst = decode_dst_reg();
88         regs[dst] = regs[dst] - regs[src];
89         break;
90     ...
91     }
92     eip += instruction_length;
93 }
94
95 3. Emulate PC's physical memory map
96
97 #define KB      1024
98 #define MB      1024*1024
99
100 #define LOW_MEMORY  640*KB
101 #define EXT_MEMORY  10*MB
102
103 uint8_t low_mem[LOW_MEMORY];
104 uint8_t ext_mem[EXT_MEMORY];
105 uint8_t bios_rom[64*KB];
106
107 uint8_t read_byte(uint32_t phys_addr) {
108     if (phys_addr < LOW_MEMORY)
109         return low_mem[phys_addr];
110     else if (phys_addr >= 960*KB && phys_addr < 1*MB)
111         return bios_rom[phys_addr - 960*KB];
112     else if (phys_addr >= 1*MB && phys_addr < 1*MB+EXT_MEMORY) {
113         return ext_mem[phys_addr-1*MB];
114     }
115     else ...
116 }
117
118 void write_byte(uint32_t phys_addr, uint8_t val) {
119     if (phys_addr < LOW_MEMORY)
120         low_mem[phys_addr] = val;
121     else if (phys_addr >= 960*KB && phys_addr < 1*MB)
122         ; /* ignore attempted write to ROM! */
123     else if (phys_addr >= 1*MB && phys_addr < 1*MB+EXT_MEMORY) {
124         ext_mem[phys_addr-1*MB] = val;
125     }
126     else ...
127 }
128

```