
CS480

Compilers

SA2 Pointer Example

SA2 Boolean Operators Example

April 11, 2011

OpCode Execution

```
void intExecute(int wOpcode, /*opcode of current quad to be executed */
               int wOperand1, /*operand1 value if necessary for opcode*/
               int wOperand2, /*operand2 value if necessary for opcode*/
               int wOperand3) /*operand3 value if necessary for opcode*/
{
    . . .
    switch (wOpcode)
    {
        case OP_ADD:          gStack[wOperand3] = wOperand1 + wOperand2;
                              break;

        . . .

        case OP_DEREFERENCE: gStack[wOperand3] = gStack[wOperand1];
                              break;
    }
}
```

Addressing Modes

```
int intDecode(int wMode,          /* mode of the operand */
              int wAddress)      /* address of Op */
{
    switch (wMode)
    {
        case IMMEDIATE:          return (wAddress); // 0

        case GLOBAL_LVALUE:     return (wAddress); // 1

        case GLOBAL_RVALUE:     return (gStack[wAddress]); // 2

        case LOCAL_LVALUE:      return (gAP + wAddress); // 3

        case LOCAL_RVALUE:      return (gStack[gAP + wAddress]); //4
    };
}
```

Pointer Example

```
int foo()  
{  
    int *pA;  
    int data = 9;  
  
    pA = &data;  
  
    *pA = 6;  
  
    return *pA;  
}
```

Pointers & Arrays

```
int c[5];
main ()
{
    int *a;
    int b[3];

    b[1] = 100;
    c[2] = 90;
    *a = c[2];
    output(b[1], *a);
}
```

Pointers & Arrays

```
int c[5];
main ()
{
    int *a;
    int b[3];

    c[2] = 1;
    b[c[2]] = 100;
    a = &b[1];
    output(c[2], *a);
}
```

Pointer as argument

```
int foo(b, c)
int b;
int *c;
{
    *c = b;
    return c;
}
```

```
main ()
{
    int x;
    int *c;
    int b[3];

    c = &x;

    b[0] = 9;
    b[1] = b[0];

    foo(b[1], c);
    output(*c, x);
    output(c);
}
```

Relational Operator

```
main ()
{
    int x;
    int y=9;

    /* store 0 into x */
    x = y < 1;
}
```


Relational Operator

```
main ()
{
    int x;
    int y=9;

    /* store 1 into x */
    x = y < 10 && y > 5 ;
}
```