CASE WESTERN RESERVE UNIVERSITY
Case School of Engineering
Department of Electrical Engineering and Computer Science
EECS 337 Systems Programming (Compiler Design)
Fall 2013
Assignment #11
30 Points + 5 points extra credit
Due: November 19, 2013

Part 1: Reading

Compilers - Principles, Techniques, & Tools, 2nd Edition, Sections 8.2, 8.3, 8.4
Homework Exercises – laboratory assignment and answer the questions at the end of the assignment.

Introduction

In this assignment you optimize the calculator program to generate improved intermediate code. You optimize the quad linked list by improving the generation and/or optimizing the linked list. You are allowed to improvement the code in any location in the compiler (scanner, symbol table, parser, IR code generator).

You can reuse temporary locations, optimize temporary variables and/or reduce the instruction set. You use the previous test programs to compare your results. For extra credit you add optimization methods for generating the array quads.

Part 2: Laboratory

From a console window, make a directory on your computer in your EECS337 directory under your Case ID and call it hw11.

```
mkdir ~/EECS337/caseid/hw11/          ; where caseid is YOUR Case ID, enter all in lower case
```

Change directory to the hw11 directory.

```
cd ~/EECS337/caseid/hw11/
```

Download a copy of: hw11_caseid.tar file to the hw11 directory from http://blackboard.case.edu/ in the EECS337 homework assignment area. To untar the tar file type the command:

```
tar xvf hw11_caseid.tar
```

The following files will be created in the current working directory. Codegen.c is included to fix a bug in the get_address routine to update the memory pointer based on the size of the input.
Makefile
hw11_test.sh

Copy the following files from the assignment 10 directory to this directory with the commands:

```
cp ../hw10/lex.l .
cp ../hw10/main.c .
cp ../hw10/quad.c .
cp ../hw10/symbol_table.c .
cp ../hw10/yacc.y .
cp ../hw10/yystype.h .
```

Optional: If you did not complete the last assignment you can download the hw10solutions.tar file from the assignment directory and use that as your starting code. Use the command “tar xvf hw10solutions.tar”. A hw10/ directory will be created with the files. In that case edit all the files and change “caseid” to your Case ID.

You are now ready to solve the laboratory assignment.

Part 3: Laboratory Assignment
Edit the files and make changes to implement code optimization. You should consider where you want to do the optimization and design and implement one change at a time. If that does not show an improvement then try another method.

You can change the code in the parser functions (quad.c) to generate better quads or write a state machine to post process the quad linked listed. One example of optimization to remove t1 is shown below.

```
       t1 = 1;
       a = t1;

       a = 1;
```

Compare the results of this program to the results of assignment 10 to see the changes and quantify the results.

diff hw11_test.txt ../hw10/hw10_test.txt | more

For example edit the quad.c file and change the new_quad1 function to remove the temporary operands when both operands are either an identifier or a constant. This checks if the two previous expressions were an identifier or constant production.

```
/*
 *     check if optimize quad
 */
       if( q1->operator == '=' && q1->dst_type == TYPE_TEMPORARY && ( q1-
>op1_type == TYPE_IDENTIFIER || q1->op1_type == TYPE_CONSTANT) &&
           q2->operator == '=' && q2->dst_type == TYPE_TEMPORARY && ( q2-
>op1_type == TYPE_IDENTIFIER || q2->op1_type == TYPE_CONSTANT))
       {
              data.temp--;
              data.temp--;
              quad1 = new_quad( operator, TYPE_TEMPORARY, next_temp(), q1-
>op1_type, q1->op1_index, q2->op1_type, q2->op1_index);
              free_quad( q1);
              free_quad( q2);
              return quad1;
       }

/*
 *     do the previous thing
 */
```

Save the quad.c file.

Build the calc program and fix any errors using the commands:
make clean
make

To test your version, type the command below:
./calc + symbol and enter the lines below.

```
./calc +symbol
for caseid start time: Mon Nov 11 11:22:12 2013
Enter calculator expression and $ to exit
int a;
int b;
int c;
a = (b + 2) * 2 + (b + 2);
```

```
      t1 = b + 2
      t1 = t1 * 2
      t2 = b + 2
      a = t1 + t2
$
symbol table:
index: 1 identifier: a length: 2 specifier: int
index: 2 constant: 4 length: 2 format: decimal
index: 3 identifier: b length: 2 specifier: int
index: 4 identifier: c length: 2 specifier: int
index: 5 constant: 2 length: 2 format: decimal
```

Test using the test files using the commands below. The output for the first file is shown.
./calc +symbol ../hw10/math24.txt
./calc +symbol ../hw10/math25.txt

```
for caseid start time: Mon Nov 11 11:41:09 2013
      c1 = 1
      s2 = c1 + 2
      i4 = s2 + 4
      t2 = s2 / i4
      t3 = t2 * 10
      l8 = c1 + t3
symbol table:
index: 1 identifier: c1 length: 3 specifier: char
index: 2 identifier: s2 length: 3 specifier: short
index: 3 constant: 2 length: 2 format: decimal
index: 4 identifier: i4 length: 3 specifier: int
index: 5 constant: 4 length: 2 format: decimal
index: 6 identifier: l8 length: 3 specifier: long
index: 7 constant: 8 length: 2 format: decimal
index: 8 constant: 1 length: 2 format: decimal
index: 9 constant: 10 length: 3 format: decimal
```

Part 4: Output Generation
When all your lab assignments have been completed execute the homework script file "./hw11_test.sh" using the command below.
./hw11_test.sh &> hw11_test.txt

Print out the hw11_test.txt file and put your name, assignment number and date on it. Turn in the file for the assignment and answer the questions at the end of the assignment. If you made a lot of changes to one file then print out that file and include it with your assignment.

Your final directory structure for the calc compiler should be as below (using your Case ID):
EECS337/caseid/hw11/Makefile
EECS337/caseid/hw11/calc
EECS337/caseid/hw11/hw11_caseid.tar
EECS337/caseid/hw11/hw11_test.sh
EECS337/caseid/hw11/hw11_test.txt
EECS337/caseid/hw11/lex.l
EECS337/caseid/hw11/main.c
EECS337/caseid/hw11/quad.c
EECS337/caseid/hw11/symbol_table.c
EECS337/caseid/hw11/yacc.y
EECS337/caseid/hw11/yystype.h

Part 5: Extra Credit (5 points)
Save a copy of your code and then edit the source files to optimize the array quads.

```
X = Y[ Z ]
X[ Y ] = Z
```

Build the calc program and fix any errors using the commands:

```
make clean
make
```

Test using the extra credit test files using the commands below:

```
./calc +symbol ../hw10/math26.txt       ; X[ Y ] = Z
```

Edit the hw11_test.sh file and remove the comment to test the math26.txt file. When the extra credit code is complete and working then execute the homework script file "./hw11_test.sh" using the command below.

```
./hw11_test.sh &> hw11_test.txt
```

Print out this version of the **hw11_test.txt** file. Mark this output with **EXTRA CREDIT** and put your name, assignment number, date on it and turn it in instead of the file from Part 4. Upload **ONLY** your quad.c file to blackboard in the assignment area. Submit it with the comments **hw11** with your **CaseID** on the title line. The extra credit will be graded on-line and your score will be added to your assignment score. Answer the questions at the end of the assignment.

Part 6: Laboratory Questions

1) Explain the type of optimization you performed in this assignment and quantify the results. (Reduction in the number of temporaries, number of quads and instructions.

	HW10 TEMPS	HW11 TEMPS	HW10 QUADS	HW11 QUADS
Math24.txt				
Math25.txt				
Math26.txt				