CS 241 – Week 3 Tutorial

Writing an Assembler Pt. I

Spring 2015

1 Summary

1. Symbol tables
2. Assembly errors
3. C++ Review

2 Problems

1. Construct the symbol table for the following MIPS assembly program.

```
begin:
label: beq $0, $0, after
jr $4

after:
sw $31, 16($0)
lis $4
abc0: abc1: .word after

loadStore:
lw $20, 4($0)
sw $20, 28($0)

end:
```

2. Identify the errors in the following assembly language program.

```
label: label: .word label
.word ; 0
.word aaaaa
.word 1 2 3
.word 2147483648 abcde:
.word ,
```

3 C++ Review

1. STL containers: pair, vector, list, map, and set
2. There are a number of flaws in the following code snippet. How can this piece of code be improved?

```
#include <iostream>
#include <vector>
#include <set>
#include <map>
#include <string>
#include <algorithm>
using namespace std;

bool foo(vector<string> v) {
    if (v.size() > 16) {
            return true;
    } else {
            return false;
        }
}

int bar(map<string, map<vector<string>, int> >  m, string w) {
        return m[w].size();
}

bool baz(string fruit) {
        return fruit == "apple" || fruit == "pear" || fruit == "mango" ||
                fruit == "coconut" || fruit == "kiwi" || fruit == "pepper";
}


string temp;
bool qux(pair<vector<string>, int> p) {
        int count = 0;
        for (int i = 0; i < p.second; ++i) {
                if (p.first[i] == temp) count++;
        }
        if (count > p.second/2) {
                return true;
        } else {
                return false;
        }
}

int main() {
        vector<string> fruits;
    map<string, map<vector<string>, int> > fruitMap;
        while (true) {
                string fruit;
            cin >> fruit;
            fruitMap[fruit][fruits] = fruits.size();

                int mode;
                if (fruit == "apple") {
                    mode = 0;
                } else if (fruit == "banana") {
                    mode = 1;
                } else if (fruit == "tangelo") {
                    mode = 2;
                } else {
                    throw 143;
                }

                fruits.push_back(fruit);
        if (foo(fruits)) {
                    cout << "Many␣fruits" << endl;
                }

        int val = bar(fruitMap, fruit);

                bool flag1 = false, flag2 = false;
                if (val > 12345) {
                    flag1 = true;
                } else if (fruits.size() > 8 && fruits.size() < 12) {
                    flag1 = true;
                } else if (mode == 1) {
                        flag2 = true;
                }

                if (flag2 || flag1 && baz(fruit)) {
                        break;
        }
        }


    for (map<string, map<vector<string>, int> >::iterator it = fruitMap.begin();
          it != fruitMap.end(); ++it) {
        temp = (*it).first;
                cout << count_if((*it).second.begin(), (*it).second.end(), qux);
        }
}
```