CSE 4130

Formal Languages & Compilers Lab

Name: Sarwar Saif

Date of submission: 25/7/2018

```
#include<bits/stdc++.h>
using namespace std;

struct token
{
   int num;
   string name, idType, dataType, scope;

} symbol[1000];

map<string,int> keyword;
int inc=0;
void showSymbolTable()
{
   for(int i=1; i<=inc; i++)
   {
      cout << symbol[i].num << "\t"
          << symbol[i].name << "\t"
          << symbol[i].idType << "\t"
          << symbol[i].dataType << "\t"
          << symbol[i].scope << endl;
   }
}

void insertTable()
{
   string name, idType, dataType, scope;
   cout << "Input name, idType, dataType, scope" <<
endl;
   cin >> name >> idType >> dataType >> scope;
   inc++;
   symbol[inc].num=inc;
   symbol[inc].name=name;
   symbol[inc].idType=idType;
   symbol[inc].dataType=dataType;
   symbol[inc].scope=scope;

}
void updateTable()
{
   int no,n;
   string name, idType, dataType, scope;
   cout << "Enter an Id number" << endl;
   cin >> no;
   cout << "Enter Number 1.name Or 2.idType Or
3.dataType Or 4.Scope\n" << endl;
   cin >> n;

   if(n==1)
   {
      cout << "Enter updated name:\n";
      cin >> name;
      symbol[n].name=name;
   }
   if(n==2)
   {
      cout << "Enter updated idType:\n";
      cin >> idType;
      symbol[n].idType=idType;
   }
   if(n==3)
   {
      cout << "Enter updated dataType:\n";
      cin >> dataType;
      symbol[n].dataType=dataType;
   }
   if(n==4)
   {
      cout << "Enter updated scope:\n";
      cin >> scope;
      symbol[n].scope=scope;
   }
   cout << "Updated Table: " <<endl;
   showSymbolTable();

}
void deleteValue()
{
   int n;
   cout << "Choose row you want to delete\n" <<
endl;
   cin >> n;

   for(int i=1; i<inc; i++)
   {
      if(i>=n && i!=inc)
      {
         symbol[i].num= symbol[i].num;
         symbol[i].name=symbol[i+1].name;
         symbol[i].idType=symbol[i+1].idType;
         symbol[i].dataType=symbol[i+1].dataType;
         symbol[i].scope=symbol[i+1].scope;
      }
   }
   inc--;
   cout << "Updated Table: " <<endl;
   showSymbolTable();
}
void searchTable()
{
   string s;
   cout << "Insert keyword you want to search" <<
endl;
   cin >> s;

   for(int i=1; i<=inc; i++)
   {
      if(s==symbol[i].name)
      {
         cout << "Found in row " << i << " and it's a
variable name." << endl;
      }
      else if(s==symbol[i].idType)
      {
         cout << "Found in row " << i << " and it's a
idType." << endl;
      }
      else if(s==symbol[i].dataType)
      {
         cout << "Found in row " << i << " and it's a
dataType." << endl;
      }
      else if(s==symbol[i].scope)
      {
         cout << "Found in row " << i << " and it's a
scope." << endl;
      }
   }
}

void step4()
{
   int cc=0;
   FILE *p1,*p2,*p3;
   char c,d;
   int flag=0,fl=0;
   p1 = fopen("input.txt","r");
   p2 = fopen("data.txt","w");
   int f=0;
   string s="";
   if(!p1) printf("\nFile can't be opened!");
   else
   {
      while((c = fgetc(p1)) != EOF)
      {
         if(c=='[')
         {
            fputc(c,p2);
            f=1;
            cout << c;
            s="";
         }
         else if(f==1 && c==' ')
         {
            if(s=="id")
            {
               cc++;
               fputc('i',p2);
               fputc('d',p2);
               fputc(' ',p2);

                  cout << s << " " << cc << "]";

            }
            if(!(s=="id"))
            {

               while((c = fgetc(p1)) != ']')
               {
                  fputc(c,p2);
                  cout << c;
               }
               fputc(c,p2);
               cout << c;
            }
            f=0;
            s="";
         }
         else
         {
            if(c!=' ')
            {
               s+=c;
            }
         }
      }
   }

   fclose(p1);
   fclose(p2);
}


int main(void)
{

   FILE *p1,*p2,*p3;
   char a,c,d;

   int flag=0,fl=0;
   p1 = fopen("input2.txt","r");
   p2 = fopen("data2.txt","w");
   int f=0,cnt=0;
```

```
string s="",s1="",s3="global";
string check[100];
if(!p1) printf("\nFile can't be opened!");
else
{
   while((c = fgetc(p1)) != EOF)
   {
      if(c==' ' || c=='\n')
      {
         fputc('$',p2);
         s="";
      }
      else
      {
         a='\n';
         if(c=='/')
         {
            c=fgetc(p1);
            if(c=='*')
               a='/';
            while((c = fgetc(p1)) != a)
            {
            }
         }
         else
         {
            if(c!='\n' || c!='\0' || c!=';')
            {

               if(c=='(')
               {
                  fputc('$',p2);
                  fputc('f',p2);
                  fputc('u',p2);
                  fputc('n',p2);
                  fputc('c',p2);
                  fputc('t',p2);
                  fputc('i',p2);
                  fputc('o',p2);
                  fputc('n',p2);
                  fputc('$',p2);
                  while((c = fgetc(p1)) != ')')
                  {
                     if(c==' ')
                     {
                        fputc('$',p2);
                     }
                     else
                     {
                        fputc(c,p2);
                     }
                  }
               }
               else
               {
                  fputc(c,p2);
               }
            }

         }
      }
   }
}
fclose(p1);
fclose(p2);
p2 = fopen("data2.txt","r");

string jk[100];
int k=0;
while((c = fgetc(p2)) != EOF)
{
   if(c!='$')
   {
      jk[k]+=c;
   }
   else
   {
      k++;
      c = fgetc(p2);
      if(c!='$')
      {
         jk[k]+=c;
      }
   }
   //cout<<jk[k]<<endl;
}
keyword["double"]=1;
keyword["int"]=1;
keyword["float"]=1;
keyword["main"]=1;
int chk=0;
string p="global";
cout << "\t\tStep 2" << endl;
cout << "\t\tSymbol Table\n";
cout << "\t\t------------\n";
for(int i=0; i<k; i++)
{
   if(jk[i]=="main" && jk[i+1]!="function")
   {
      cout << ++inc << "\t" << jk[i+1] << "\tfunc\t"
<< jk[i] << "\tglobal" << endl;
      symbol[inc].num=inc;
      symbol[inc].name=jk[i+1];
      symbol[inc].idType="func";
      symbol[inc].dataType=jk[i];
      symbol[inc].scope="global";
```

```
         p="main";


      }
      else if(keyword[jk[i]]==1)
      {
         if(jk[i+2]=="function" && jk[i+1]!="function")
         {
            if(jk[i+1]=="main"){
               cout << ++inc << "\t" << jk[i+1] <<
"\tfunc\t" << jk[i] << "\t" << "global" << endl;
               symbol[inc].num=inc;
               symbol[inc].name=jk[i+1];
               symbol[inc].idType="func";
               symbol[inc].dataType=jk[i];
               symbol[inc].scope="global";
               p=jk[i+1];

            }
            else{
               cout << ++inc << "\t" << jk[i+1] <<
"\tfunc\t" << jk[i] << "\t" << p << endl;
               symbol[inc].num=inc;
               symbol[inc].name=jk[i+1];
               symbol[inc].idType="func";
               symbol[inc].dataType=jk[i];
               symbol[inc].scope=p;
               p=jk[i+1];

            }
            //cout<<"2nd IF"<<endl;

         }
         else if(jk[i+1]!="function")
         {
            cout << ++inc << "\t" << jk[i+1] << "\tvar\t"
<< jk[i] << "\t" << p << endl;
            symbol[inc].num=inc;
            symbol[inc].name=jk[i+1];
            symbol[inc].idType="var";
            symbol[inc].dataType=jk[i];
            symbol[inc].scope=p;
         }
      }
   }
   cout << "Step 3\nOperations in symbol table:\n"
<< endl;
   cout <<
"Choose:\n1.Insert()\n2.update()\n3.delete()\n4.sea
rch()\n5.display()\n6.exit\n";
   int tt;
```

```
   while(cin >> tt && tt!=6)
   {
      if(tt==1)
         insertTable();
      if(tt==2)
         updateTable();
      if(tt==3)
         deleteValue();
      if(tt==4)
         searchTable();
      if(tt==5)
         showSymbolTable();
      cout <<
"Choose:\n1.Insert()\n2.update()\n3.delete()\n4.sea
rch()\n5.display()\n6.exit\n";


   }
   cout << "Step 4: " << endl;
   step4();

   return 0;
}
```