

Tarea de Laboratorio 2

Christofer Fabián Chávez Carazas

Universidad Nacional de San Agustín de Arequipa

Escuela Profesional de Ciencia de la Computación

Compiladores

12 de octubre de 2017

El objetivo de la práctica era crear un pequeño compilador que traduzca un pseudocódigo a lenguaje c++. El pseudocódigo es el siguiente:

```
Inicio
variables a,b,c:entero
Leer a
Leer b
c<-a+b
Escribir c
```

El programa es el siguiente:

```
#include <iostream>
#include <vector>
#include <cstdio>

using namespace std;

enum Errores {TIPE, ASIG, NOEXVAR, GENERERROR};

class Error{
public:
    Error(int a, int b, string s){
        numLinea = a;
        error = b;
        linea = s;
    }
    int numLinea;
    int error;
    string linea;
};

int IDfind(vector<string> &v, string s){
    for(int i = 0; i < v.size(); i++){
        if(v[i] == s) return i;
    }
    return -1;
}

bool find(vector<string> &v, string s){
    if(IDfind(v,s) == -1) return false;
    return true;
}

int main(int argc, char const *argv[]){
    string line = "";
    int estado = 0;
    int numLinea = 0;
    vector<string> idVariables;
    vector<string> tipoVariables;
    try{
        while(true){
            cin>>line;
```

```

numLinea++;
if(estado == 0 and line == "Inicio"){
    cout<<"#include <iostream>"<<endl<<"using namespace std;"<<endl<<"int main(int argc, ←
        char const *argv[]) {"<<endl;
    estado = 1;
}
else if(estado == 1 and line == "variables") estado = 2;
else if(estado == 2){
    string temp = "";
    int num = 0;
    for(char c : line){
        if(c == ',' or c == ':'){
            num++;
            idVariables.push_back(temp);
            temp.clear();
        }
        else temp.push_back(c);
    }
    if(temp != "entero" and temp != "real"){
        throw(Error(numLinea, TIPE, line));
    }
    for(int i = 0; i < num; i++){
        tipoVariables.push_back(temp);
    }
    for(int i = 0; i < idVariables.size(); i++){
        string tipo = "";
        if(tipoVariables[i] == "entero") tipo = "int";
        else if(tipoVariables[i] == "real") tipo = "float";
        cout<<tipo<<" "<<idVariables[i]<<" "<<endl;
    }
    estado = 3;
}
else if(estado == 3){
    if(line == "Leer") estado = 4;
    else{
        string temp = "";
        string uno = "";
        string dos = "";
        string tres = "";
        int subestado = 0;
        for(char c : line){
            if(subestado == 0 and c == '<'){
                if(!find(idVariables, temp)) throw(Error(numLinea, NOEXVAR, line));
                uno = temp;
                temp.clear();
                subestado = 1;
            }
            else if(subestado == 1){
                if(c != '-'){
                    throw(Error(numLinea, ASIG, line));
                }
            }
            else{
                //temp.push_back(c);
                subestado = 2;
            }
        }
        else if(subestado == 2 and c == '+'){
            if(!find(idVariables, temp)) throw(Error(numLinea, NOEXVAR, line));
            dos = temp;
            temp.clear();
            subestado = 3;
        }
        else temp.push_back(c);
        if(!find(idVariables, temp)) throw(Error(numLinea, NOEXVAR, line));
        tres = temp;
        cout<<uno<<"="<<dos<<"+"<<tres<<" "<<endl;
        estado = 5;
    }
}
else if(estado == 4){
    if(!find(idVariables, line)) throw(Error(numLinea, NOEXVAR, line));
    cout<<"cin>>"<<line<<" "<<endl;
    estado = 3;
}
else if(estado == 5 and line == "Escribir") estado = 6;
else if(estado == 6){
    if(!find(idVariables, line)) throw(Error(numLinea, NOEXVAR, line));
    cout<<"cout<<"<<line<<"<<endl;"<<endl;
    estado = 7;
}
else if(estado == 7){
    cout<<"return 0;"<<endl<<"}"<<endl;
    break;
}
else throw(Error(numLinea, GENERERROR, line));
}
}
catch(Error e){
    fprintf(stderr, "Line %d:%s\n", e.numLinea, e.linea.c_str());
    switch(e.error){
        case TIPE:
            fprintf(stderr, "No existe el tipo\n");
            break;

```

```

        case ASIG:
            fprintf(stderr, "Error en la operacion asignacion\n");
            break;
        case NOEXVAR:
            fprintf(stderr, "No existe la variable\n");
            break;
        case GENERERROR:
            fprintf(stderr, "ERROR\n");
            break;
    }
}
return 0;
}

```

El compilador va generando el código intermedio mientras va leyendo el pseudocódigo. El programa cuenta con un contador de líneas para un mejor manejo de los errores. Los identificadores son guardados en el vector *idVariables*, mientras que sus respectivos tipos son guardados en el vector *tipoVariables*. El compilador reconoce variables de tipo entero y real, y cada vez que se necesita, realiza una búsqueda en el vector *idVariables* para verificar que el identificador no esté repetido o que sí exista.

Resultados:

- Con el pseudocódigo sin alterar:

```

xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ cat pseudo
Inicio
variables a,b,c:entero
Leer a
Leer b
c<-a+b
Escribir c
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ ./run < pseudo > temp.cpp
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ cat temp.cpp
#include <iostream>
using namespace std;
int main(int argc, char const *argv[]){
    int a;
    int b;
    int c;
    cin>>a;
    cin>>b;
    c=a+b;
    cout<<c<<endl;
    return 0;
}
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ g++ temp.cpp -o run2
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ ./run2
1
2
3
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ █

```

- Cambiando el tipo entero por real:

```
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ cat pseudo2
Inicio
variables a,b,c:real
Leer a
Leer b
c<-a+b
Escribir c
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ ./run < pseudo2 > temp.cpp
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ cat temp.cpp
#include <iostream>
using namespace std;
int main(int argc, char const *argv[]){
float a;
float b;
float c;
cin>>a;
cin>>b;
c=a+b;
cout<<c<<endl;
return 0;
}
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ g++ temp.cpp -o run2
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ ./run2
1
2
3
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ ./run2
1.5
2.7
4.2
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ █
```

- Leyendo una variable que no ha sido definida:

```
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ cat pseudo3
Inicio
variables a,b,c:real
Leer e
Leer b
c<-a+b
Escribir c
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ ./run < pseudo3 > temp.cpp
Line 5:e
No existe la variable
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ █
```

- Poniendo un tipo que no es reconocido:

```
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ cat pseudo4
Inicio
variables a,b,c:short
Leer a
Leer b
c<-a+b
Escribir c
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ ./run
documento/ main.cpp  pseudo  pseudo2  pseudo3  pseudo4  run      run2      temp.cpp  tt.cpp
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ ./run < pseudo4 > temp.cpp
Line 3:a,b,c:short
No existe el tipo
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/UNSACS/Compiladores/Lab/Tarea2$ █
```