

SWE4001 – System Programming
Lab – 8
Direct Linking Loader Pass-2

Write a C program to Implement pass two of a direct-linking loader. Sample three input files are given.
Sample output files are given.

Code:

```
#include<bits/stdc++.h>
using namespace std;
class Pass2{
int PROGADDR;
int CSADDR;
string PROGRAM_NAME;
int LENGTH;
vector<string> files;
map<string,int> controlSection;
map<string,int> ESTAB;
string WHITESPACE = " \n\r\t\f\v";
string ltrim(const string &s)
{
size_t start = s.find_first_not_of(WHITESPACE);
return (start == string::npos) ? "" : s.substr(start);
}
string rtrim(const string &s)
{
size_t end = s.find_last_not_of(WHITESPACE);
return (end == string::npos) ? "" : s.substr(0, end + 1);
}
string trim(const string &s) {
return rtrim(ltrim(s));
}
void writeLine(vector<string> line,bool isPass1=true){
string record = line[0];
if(isPass1)
{ fstream outputFile("output.txt",ios::app);
if(record=="H"){
CSADDR = PROGADDR;
outputFile<<line[1]<<"\t\t\t"<<hex<<PROGADDR<<"\t"<<line[3]<<endl;
// add ControlSection in memory;
controlSection.insert({line[1],PROGADDR});
PROGRAM_NAME = line[1];
LENGTH = stoi(line[3],nullptr,16);
PROGADDR+=stoi(line[3],nullptr,16);
}
else if(record=="D"){
for(int i=1;i<line.size()-1;i+=2){
ESTAB.insert({line[i],CSADDR+stoi(line[i+1],nullptr,16)});
outputFile<<"\t\t"<<line[i]<<"\t"<<hex<<CSADDR+stoi(line[i+1],
nullptr,16)<<endl;
}
}
```


```

}}else{
fstream out("pass2Out.txt", ios::app);
int index = 1;
if(record=="H"){
out<<index++<<"\t"<<line[1]<<"\t"<<hex<<controlSection[line[1]
]<<endl;
}
else if(record=="R"){
for(int i=1; i<line.size();i++){
cout<<line[i]<<"===="<<ESTAB[trim(line[i])]<<endl;
out<<index++<<"\t"<<line[i]<<"\t"<<hex<<ESTAB[trim(line[i])]<<endl;
}
out<<"\n\n\n\n";
}
}
}
public:
Pass2(int proAddr, vector<string> f){
PROGADDR = proAddr;
CSADDR = proAddr;
files = f;
}
void generatePass2(){
for(int i=0;i<files.size();i++){
fstream inputFile(files[i],ios::in);
if(inputFile.is_open()){
string lineBuffer;
while(getline(inputFile,lineBuffer)){
stringstream ss(lineBuffer);
string pointerBuffer;
vector<string> line;
while(getline(ss, pointerBuffer, '^')){
line.push_back(pointerBuffer);
}
writeLine(line,false);
}
}
}
}
void doPass1(){
for(int i=0;i<files.size();i++){
fstream inputFile(files[i],ios::in);
if(inputFile.is_open()){
string lineBuffer;
while(getline(inputFile,lineBuffer)){
stringstream ss(lineBuffer);
string pointerBuffer;
vector<string> line;
while(getline(ss, pointerBuffer, '^')){
line.push_back(pointerBuffer);
}
writeLine(line);
}
}
}
}
}

```

```
};
int main(){
vector<string> files = {"proga.txt","progb.txt","progc.txt"};
Pass2 p2(16384, files);
p2.doPass1();
p2.generatePass2();
}
```

Output:

 output.txt

1	PROGA		4000	0063
2		LISTA	4040	
3		ENDA	4054	
4	PROGB		4063	007F
5		LISTB	40c3	
6		ENDB	40d3	
7	PROGC		40e2	0051
8		LISTC	4112	
9		ENDC	4124	
10	PROGA		4000	0063
11		LISTA	4040	
12		ENDA	4054	
13	PROGB		4063	007F
14		LISTB	40c3	
15		ENDB	40d3	
16	PROGC		40e2	0051
17		LISTC	4112	
18		ENDC	4124	
19				



pass2Out.txt

1	1	PROGA	4000
2	1	LISTB	40c3
3	2	ENDB	40d3
4	3	LISTC	4112
5	4	ENDC	4124
6			
7			
8			
9			
10	1	PROGB	4063
11	1	LISTA	4040
12	2	ENDA	4054
13	3	LISTC	4112
14	4	ENDC	4124
15			
16			
17			
18			
19	1	PROGC	40e2
20	1	LISTA	4040
21	2	ENDA	4054
22	3	LISTB	40c3
23	4	ENDB	40d3
24			
25			