



Construção de Compiladores

Varredura de Lexemas

Professor: Luciano Ferreira Silva, Dr.



Leitura de arquivos com C++

■ Abrir um arquivo externo:

- ✓ Deve-se usar a biblioteca de classes *fstream*:

```
#include <fstream>
```

- ✓ Usa-se a classe *ifstream*. Exemplo:

```
ifstream arq("leaq.cpp");
```

OU

```
ifstream arq;
```

...

```
arq.open("leaq.cpp");
```



Leitura de arquivos com C++

■ Ler um arquivo aberto:

✓ Pode-se usar o operador >> ou as funções get ou getline;

✓ Exemplo:

```
#include <fstream>
```

```
//...
```

```
ifstream arq;
```

```
char ch;
```

```
//...
```

```
arq.open(...);
```

```
while (! arq.eof) {
```

```
    arq >> ch;
```

```
}
```



Leitura de arquivos com C++

■ Observação:

- ✓ O operador de leitura `>>` por padrão ignora os chamados “espaço em branco” como:
 - o espaço (`' '`),
 - o caractere de tabulação (`'\t'`),
 - o caractere de mudança de linha (`'\n'`);
- ✓ Para alterar este padrão usa-se o comando:
`arq.unsetf(ios::skipws);`



Leitura de arquivos com C++

- ✓ A função `get` lê todos os caracteres de um arquivo;
- ✓ Exemplo: `arq.get(ch);`
- ✓ A função `getline` lê uma quantidade pré-determinada de caracteres de uma arquivo;
- ✓ Exemplo: `arq.getline(ch, 15);`



Um simples separador de lexemas com C++

```
#include <cstdlib>
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int main(int argc, char *argv[])
{
    ifstream arq;
    string lexema;
```



Um simples separador de lexemas com C++

```
arq.open ("Texto.txt");

while (!arq.eof()) {
    arq>> lexema;
    cout<< lexema <<" : ";
}

cout<<endl;

system ("PAUSE");
return EXIT_SUCCESS;
}
```




Um simples separador de lexemas com C++

```
#include <cstdlib>
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int main(int argc, char *argv[])
{
    cout<<endl;
    char string1[] = "O meu guarda-chuva
esta em casa, hoje";
```




Um simples separador de lexemas com C++

```
char * character_pointer;
character_pointer = strtok (string1, "
, . -");
while (character_pointer != NULL)
{
    cout<<character_pointer<<" : ";
    character_pointer = strtok
(NULL, " , . -");
}
cout<<endl;
system("PAUSE");
return EXIT_SUCCESS;
}
```



2º Trabalho

- Construir uma Classe ou Função que realize a varredura de lexemas o mais completa possível. Este programa deve armazenar as strings dos lexemas encontrados em uma estrutura de dados adequada, como uma tabela.

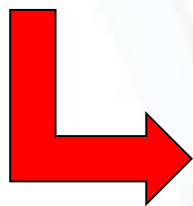


2º Trabalho

- Seu separador de lexemas deve conseguir separar e arquivar expressões como:

- `x = y + 5;`

- `int z;`



| LEXEMA | RÓTULO | COD |
|--------|--------|-----|
| x | ID | 1 |
| = | = | 2 |
| y | ID | 3 |
| + | + | 4 |
| 5 | NUM | 5 |
| ; | \$ | 6 |
| int | PR | 7 |
| z | ID | 8 |
| ; | \$ | 9 |



2º Trabalho

- Você deve entregar: o separador de lexema (a linguagem de programa é de escolha sua) e o relatório de construção do programa, abordado o desenvolvimento e alguns casos de testes



Dúvidas

UFRR – Departamento de Ciência da Computação
Construção de Compiladores – Prof. Dr. Luciano F. Silva

