



Universidade Federal de Mato Grosso do Sul
Campus de Ponta Porã
Curso: Ciência da Computação
Disciplina: Compiladores

Trabalho de Compiladores (Etapa 1 – 3,0 pontos)

Data da entrega: 21/08/2019

Especificação da Etapa 1: Implementar um analisador léxico manualmente ou com auxílio do Flex.

Convenções da linguagem para a qual o compilador será construído:

- 1) Palavras-chaves (todas são palavras reservadas e devem ser escritas com letras minúsculas):
if else int void return while
- 2) Símbolos especiais:
+ - * / < <= > >= == != = ; , () [] { } /* */
- 3) Um identificador *ID* e um número *NUM* serão definidos como:
letra → a | .. | z | A | .. | Z
dígito → 0 | .. | 9
ID → letra letra*
NUM → dígito dígito*
- 4) Os espaços em branco são compostos por brancos, mudanças de linha e tabulações. Com exceção dos espaços em branco que separam identificadores, números e palavras-chaves, todos os outros devem ser ignorados.
- 5) Os comentários devem ser precedidos pelo símbolo especial */** e sucedidos pelo símbolo especial **/*. Os comentários podem incluir mais de uma linha e podem ocorrer em qualquer lugar onde um espaço em branco possa aparecer, ou seja, os comentários não podem ocorrer no meio de identificadores, números e palavras-chaves.

Erros léxicos: caso o analisador léxico encontre símbolos ou construções que não façam parte da linguagem, deverá ser emitida uma mensagem de erro. Ainda assim, o seu analisador deve continuar o processamento.

OBSERVAÇÃO: Com a finalidade de testar o seu analisador léxico, a entrada será um arquivo-texto e a saída deverá ser uma lista com os nomes dos tokens.

Exemplo de entrada:

```
void main(void)
{ int x; int y;
  int z;
  x = 5;
  y = 10;
}
```

Saída: void, ID, (, void,), {, int, ID, ;, int, ID, ;, int, ID, ;, ID, =, NUM, ;, ID, =, NUM, ;, }.