

# Universidade Federal de Mato Grosso do Sul – Campus de Ponta Porã

*Neemias B. Silva; Vinícius Nunes Martins; Nádia Boeira*

21 de Agosto de 2019

O objetivo deste trabalho foi implementar um analisador léxico para ser avaliado como nota da primeira etapa do trabalho da disciplina de compiladores. Antes de explicar cada um dos arquivos em anexo, é bom salientar que a linguagem de programação utilizada junto com a ferramenta **Lex Yacc** foi a linguagem **c**.

Os arquivos em anexo funcionam da seguinte forma:

- **trabalho-1.l**: é o arquivo flex onde foi criado e definido as *expressões regulares* para reconhecer os *tokens* que está sendo pedido na descrição. (Ex.: *while*, *ID*, etc...) [1]
- **trabalho-1.c**: é o arquivo resultante da compilação da ferramenta **flex**. O comando pode ser feito da seguinte forma utilizando o terminal (GNU/UNIX):
  - **flex -o trabalho-1.c trabalho-1.l**
- **trabalho-1**: este executável é um arquivo resultante da compilação do compilador **gcc**; para gerar este executável é necessário digitar o seguinte comando:
  - **gcc -o trabalho-1 trabalho-1.c**

Para verificar se o nosso analisador léxico estava funcionando de acordo com o que o trabalho estava especificando foi realizado a execução de três casos de teste denominados **case\_01.in**, **case\_02.in** e **case\_03.in**. Logo abaixo, há uma descrição dos respectivos arquivos:

- **case\_01.in**: Neste arquivo foi passado o exemplo que está na descrição do trabalho.
- **case\_02.in**: Neste arquivo foi passado um exemplo que chama uma função para calcular a sequência de *fibbonaci* recursivamente;
- **case\_03.in**: Neste arquivo foi colocado vários *tokens* que não podem ser reconhecidos pela linguagem propósta para o trabalho.

## Referências

- [1] Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. *Compilers: Principles, Techniques, and Tools (2Nd Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.