COMPILER DESIGN LAB [WEEK-2] 26-08-2022

# IMPLEMENTING THE STRUCTURE SYMBOL TABLe

## **ABSTRACT**

The symbol table is a structure and a part of the compiler which maintains the data of variables, function’s names, and its datatypes with respective sizes in the code which was written by the user. This symbol table is mainly used to store all the identifiers that were generated from the lexical analysis. In this lab experiment, we are going to implement the structure of the symbol table in python with the dataframe(pandas) representation with some variables that are taken as an example.

## **IMPLEMENTATION**

* The symbol table can be created by various data structures like lists, linked lists, binary search, etc. Here the structure was implemented with the help of dictionaries in python, and the output was displayed in the form of a table with the help of pandas.Dataframe.
* Pseudo code:

**Import pandas**

//the datatypes are always keywords with respective compiler and user.

**print (“the example datatypes that should be given by the user “).**

**Symbol table = {datatypes: [keyword-1, keyword-2, keyword-3, ……, keyword-n],**

**Identifiers: [variable-1, variable-2……., variable-n]}**

**Data = pandas.Dataframe(Symbol table)**

## **PROGRAMME**

import pandas as pd

print("""int a = 588,arr[588];\nfloat decimal==588.088;\nchar letter = "h";\ndouble big\_num=123456789;

""")

symbol\_table={

"data type":["int","float","char","double","int"],

"identifiers":["a","decimal","letter","big\_num"," arr[size = 588]"],

}

df = pd.DataFrame(symbol\_table,index={1,2,3,4,5}

## **CONCLUSION**

Form this code we can conclude the output as a structured table with the 2 fields named identifiers and datatypes,

Table

Description automatically generated

Structure of symbol table

User defined example datatypes