Presentation Script

Topics : Compiler Design , Computation

Group : **Nopa Islam & Sukannya Saha**

**Slide : 1**

Good After- Noon everyone ,welcome to our group presentation. Our tentative topics is Compiler Design , Computation

**Slide : 2**

I am Nopa Islam staring the presentation , with my fellow team mate Sukannya Saha. So Lets log in to the presentation.

**Slide : 3**

Compiler is mainly one kind of program that normally converts instructions into machine code.

Normally it handle 5 types of code , they are

Source Code => It is mainly the collection of instructions

Object Code => It means sequence of statements

Byte Code => Executed by parsing and directly executing the instructions

Machine Code => Executed by CPU

Micro Code => Lowest specified level of processor and machine instructions set

**Slide : 4**

Let’s look at the flow , so compiler takes a source program as input ,generates all possible error reports ,if not then takes the input in the .exe and produces result.

Slide : 5

Here we can see, compiler is working on 6 steps. Each of them are well defined. The front end is well maintained. The backend does all the calculation

**Slide : 6**

So preprocessor modifies the input, mainly deletes all spaces . Compiler does all the critical operation in 2 parts they are (Analysis & Synthesis ). Assembler just converts the low level code to machine code. Linker + loader supplies relocate –able object code.

Memory holds the output.

My part is done now ; I wanna call My Team Mate Sukannya Saha to continue the presentation. Over to you Sukannya

Thanks Nopa for giving me the scope.

So I will start from Slide number 7

**Slide : 7**

Here we are using Symbol table as data structure .

Using Grammar to make tokens and then install it in the symbol table. All phases are connected with the table.

**Slide : 8**

So making token then making , parse tree. Then making sure it is error free. Intermediate representation passes the info to the next steps to produce the final output**.**

**Slide : 9**

Here we can see , how compiler evaluates an expression in every step . Which depends on symbol table . The flow is sequential .

**Slide : 10**

We are designing a compiler, so in our computation , error can occur . Normally 4 types of error can be found .

Lexical error may missed for detection. Syntactical error is caused by wrong syntax . Semantic error occurs for wrong types.

**Slide : 11**

In order to recover from error we can follow these strategy

Panic mode does shortest amount of change. Statement mode does prediction. Global correction looks for closet match

Abstract Syntax Tree : It is a special kind of data structures . AST does not include inessential punctuation and delimiters .

**Slide : 12**

Any question from the audience ?

**Slide : 13**

So it’s time to log out from the presentation. Thanks my team mate and all for your Cooperation. Have a good day .