SYSTEM PROGRAMMING COMPILER CONSTRUCTION (SPCC) OR COMPILER

LAST MOMENT TUITIONS

INDEX

First and Follow

Notes

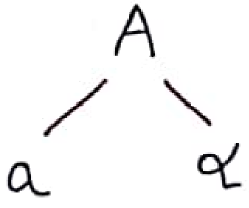
Note: Example pe direct jane se pehle hum first and follow ka basic concept samajhte hai.

Imp note: Kisi bhi start symbol ka follow \$ (dollar) pehle se hota hai by default.

First me 3 type ke cases hote hai

case 1

$A \rightarrow a\alpha$



$\therefore$ First(A) = $\{a\}$

ab α kuch bhi ho hame usse koi matlab nahi bcoz hume first direct mil gaya jo hai 'a'.

Example: $A \rightarrow aBcde$

$first(A) = \{a\}$

I hope case 1 app samajh gaye

Case 2

$A \rightarrow \epsilon$

$first(A) = \{\epsilon\}$

(Agar koi non-terminal (A) direct ∈ (epsilon) produce kar raha hai toh first of that will be epsilon)

Case 3

$A \rightarrow B$

$B \rightarrow C$

$C \rightarrow id$

Ab jaise app dekh sakte ho A ka first B hai jo ek non-terminal so mai usko first me nahi likh sakta toh hum B ka first ~~chee~~ check karte hai jo C hai non-terminal toh ap C ka first which is id yeh terminal hai isko hum first me likh sakte hai

Hence $first(A) = \{id\}$ //

Let Solve a problem:

Find First of given grammer

$A \rightarrow a$

$B \rightarrow (C) \mid id$

$C \rightarrow *T \mid \epsilon$

$\text{first}(A) = \{a\}$

$\text{first}(B) = \{(, id\}$

$\text{first}(C) = \{*, \epsilon\}$

This was the basic example and basic concept of first 🙂

SPCC (System Programming Compiler Construction) Basics:

Languages: Languages are used for communication with the system. There are also some human languages like English, Punjabi and Hindi those are used for communicating with the other humans. So there are so many languages developed those are used for communication with the humans. The languages those are human's own language are also known as Natural Languages because they are used for humans only. Like this way when a user wants to perform any operation then a user must have to use some program or some request. And the system will the respond to the request of the user. For making a request to the system, a user can use any programs, and the programs are the collection of statements and these statements are written into some Programming Languages. All the programming languages also developed in the form of some generations those are as explained below:

Machine Language or 1st GL: This is also called as the first generation computer languages. The machine language programs contain all the instructions in the binary form which are easily understandable to the computer system. So that it is very Last Moment Tuitions difficult to understand for the user because all the instructions are written into the form of 0 and 1. Basically this language was used for writing the machine instructions means how the registers will work and how they will be work for performing the requests of the users. So that we can also say that these languages are used for internal structure of the computer system. The main advantage is that this is very easy for the computer and this will also increase the speed of processing of the computer system. And the main limitation is that this will consume lots of time of user, if a user wants to understand any program and this is very difficult to learn. But the programs those are written into the machine languages are never need to translate into the other languages.

(Computer ko sirf machine language samjhti hai aur kuch nahi jo ki binary form me hoti hai 0's and 1's me ab yeh computer ke liye bahut achi hai par users ke liye bahut mushkil isilye picture me aya second Generation)

Assembly Language 2 GL: The program which is written into the machine language are very difficult to understand for the user so that there is the development of the new programming language which is also known as the second generation language which is also known as the assembly language. In the assembly language there are many mnemonics those are also called as the reserve words those are easy to understand to the users and those have some specific meaning. But the words are not machine language oriented means the words are just the English language forms so that they are not easily understood by the machine and it is needed to convert the words of the assembly language into the machine language. For converting the assembly language into the machine language, the assembler is used. This converts the whole program which is written into the assembly language into the machine language. For example, if a user wants to add two numbers then he has to use some mnemonics like this ADD A, B in this ADD is the mnemonics which is used for performing the addition operation. This makes very easy for the user to understand. The main limitation is that the program which is written into the assembly language must be converted into the machine language because the program will contain some words those are in the human language form and they are not machine language form.

(apne jaise bando ko binary language bahut hard padh rahi thi so ek lazy bande ne bola bhai meko yeah nahi jamta 0 and 1 so usne nikali binary language but hame Last MomentTuitions assembly language ko machine language me convert karna padhta hai aur uske liye assembler cahiye) assembly language ko machine language me convert karna padhta hai aur uske liye assembler cahiye)

3. **3 GL or some high level languages:** Due to the development of the assembly language and the mnemonics there is also the new development of some high level languages those are used for making the programs. There are many high level programming languages has developed at that time which contains the Java, C, C++, etc. All the high level programming languages are user friendly means the syntax of theses languages is quite simple because they contains all the words those are in the form of English language but the main problem is that they are not machine oriented means the program which is written into the high level languages is not understand by the computer so that the program must be converted into the machine language and the program which is written into the human language will cause the computer in slow of processing. Typically, a Compiler is needed to convert HLL into machine Language.

(ab ek banda bolta hai yaar meko na assembly languagr bhi hard padhti hai so usne nikali high level lnguage jo ki thori bahut natural language ki tarah hai(eg:C,C++, java,etc)

MACROS:(ab ek banda bolta hai meko na high level language bhi hard padhi hai usme kabhi kabhi agar ek cheez 500 times likhni padhe toh obvious hume hard padengi so usne nikale macros matlab voh same sentence ko uper program ke. ek baar likhdo aur ek word se replace kardo ab jaha bhi voh sentence aata hai vaha sirf voh word likhne ka)

Eg:

.

.

.

A data1

A data2

A data3

.

.

A data1

A data2

A data3

.

.

.

A data1

A data2

A data3

.

.

.

(jaise app dekh sakte ho ek hi cheez 3 baar likhni pad rahi hai program me so hum macros use karenge)

Macros

INCR <- (woh word jisse hum sab replace karenge pure sentence ko)

A data1

A data2

A data3

MEND

.

.

.

INCR

.

.

INCR

.

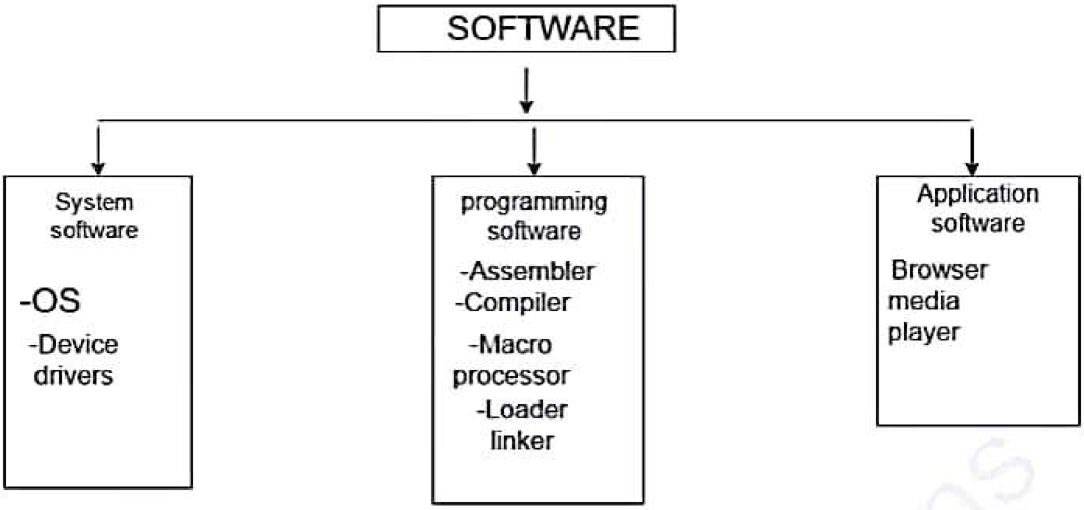
.

INCR

.

(jaise app dekh sakte hai abhi baar baar same sentence likhne ki jaroorat nahi sirf ek word likhna hai aur voh pura sentence usse replace ho jayenga)

Software and Its types



System software

Computer system divides the software systems into three major categories:

1. System Software: System software coordinates the activities and functions of hardware and software, and it controls the operations of computer hardware. It consists of variety of programs that supports the operation of a computer. System software is computer software that provides the infrastructure over which programs can operate. This makes it possible for the users to focus on an application or other problems to be solved, without needing to know the details of how machine works internally.

2. System Program Software/ Programming Software: System programs are the set of programs that are required for the effective execution of the application programs. It usually provides tools to assist programmer in writing computer programs, and software using different programming languages. The programming software tools include: ⏱ Compiler: Compiler is program that takes as input a program written in its source language (usually high level language like C++) and produces an equivalent program written in its target language (object code or machine code)

🕐 Assembler: An assembler is a language translator for the assembly language of a particular computer. Input to an assembler is an assembly language program. Output is an object program plus information that enables the loader to prepare the object program for execution.

🕐 Linkers: Both the compiler and assemblers rely on linker which a program that collects code separately compiled or assembled into a file that is directly executable. Linker also connects an object program to the code for standard library functions and to resources supplied by the OS of the computer, such as memory allocators and I/O devices.

🕐 Loaders: Compiler, assembler or linker will produce a code that is not yet completely fixed or ready to execute, but whose principal memory references are all made relative to an undetermined starting location that can be anywhere in the memory. Such code is said to be relocatable and a loader will resolve all relocatable addresses relative to a give base or starting address. In short, a loader is a program that places programs into memory and prepares them for execution. There are various loading schemes: absolute, relocating, and direct-linking. In general, the loader must load, relocate and link the object program.

🕐 Macro Processor: A macro call is an abbreviation (or name) for some code. A macro definition is a sequence of code that has a name (macro call). A macro processor is a program that substitutes and specializes macro definitions for macro calls.

🕐 Preprocessors: A pre-processor is a separate program that is called by the compiler before actual translation begins. Such pre-processor can delete comments, include other files and perform macro substitutions.

Module 01: System Software

⏻ Editors: Compilers usually accept source programs written using an editor that will produce a standard file such as ASCII file. Compilers normally are bundled together with the editors and other programs into an Interactive Development Environment or IDE.

⏻ Debuggers: Debuggers are program that can be used to determine execution errors in a compiled program. It is also packaged with a compiler in an IDE. Running a program with debugger differs from straight execution in that the debugger keeps track of most or all source code information, such as line numbers names of variable and procedures. It can also halt execution at pre-specified locations called breakpoints as well as provide information on what functions have been called and what the current values of the variables are.

⏻ Profilers: Profiler is a program that collects statistics on the behaviour of an object program during execution. Statistics that may be of interest to the programmer are the number of times each procedure is called and the percentage of execution time spent in each procedure.

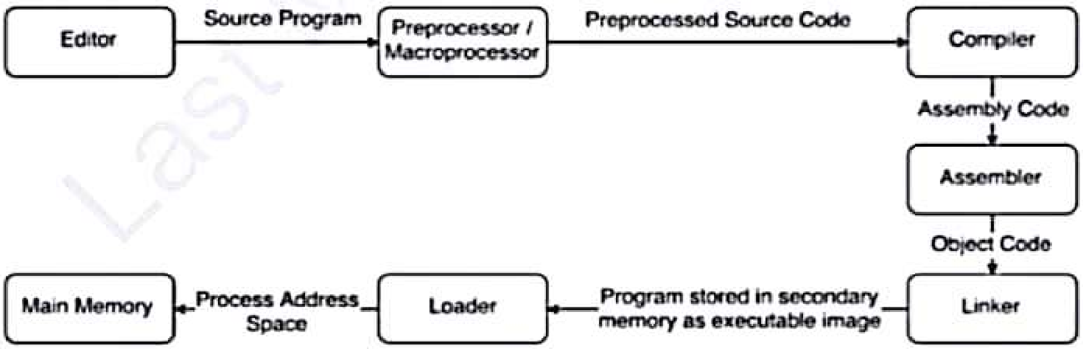
⏻ Project Managers: Modern software projects are usually so large that they are undertaken by groups of programmers rather than a single programmer. In such cases, it is important that the files being worked on by different people are coordinated and this is the job of a project of a project manager program. Project manager should coordinates the merging of separate version of the same file being produced by different programmers.

⏱ Decompiler: Decompiler is the name given to a computer program that performs the reverse operation of that of compiler i.e. it translates a file containing computer readable information to human readable. It can be used for the recovery of lost source code, computer security, interoperability and error correction. The success of decompilation depends on the amount of information present in the code being decompiled. The byte code formats used by many virtual machines often include extensive metadata and highlevel features that make decompilation.

⏱ Disassembler: Disassembler is a computer program that translates machine language into assembly language.

⏱ Interpreter: An interpreter translates high level instructions into an intermediate form. An interpreter may be program that either ▪ Executes the source code directly ▪ Translates source code into some efficient intermediate representation and executes them. E.g.: Interpreter of LISP, BASIC, etc

Lets see how the flow the works

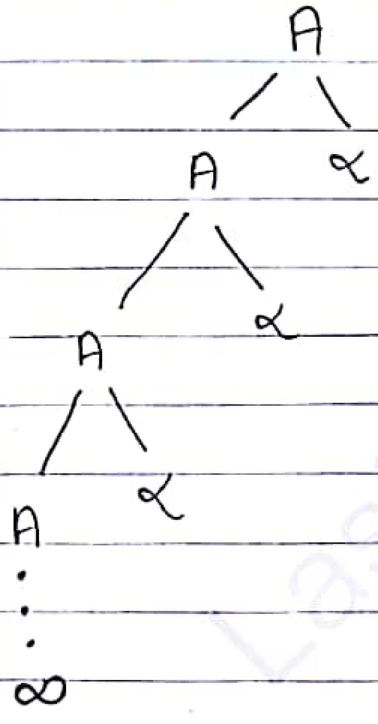


Relative Locations of System Programming Software

3. Application Software: Application software is any tool that functions and is operated by means of a computer and allow end users to accomplish one or more specific tasks. Application Software are the most common programs that run in the foreground of the computer. They tend to perform useful tasks which are not associated with computer maintenance, system boot-up, or hardware communication. Application software is directly reliant on the Systems Software to communicate to the physical components of the computer and cannot operate without it. Application Software are the most familiar forms of software and come in a variety of types. Most often they can be accessed through the graphical user interface of the operating system being used by double-clicking on an icon. Typically, application software includes: 🕒 Industrial Automation 🕒 Office Suites 🕒 Databases 🕒 Media Players 🕒 Computer Games 🕒 Photo-editing programs 🕒 Accounting programs

Elimination of left recursion

if $A \to A\alpha | \beta$ then it contains left recursion



matlab A na khud ko hi baar baar call karta hai aur infinite loop me jata hai

↓

```
A( )
{
 A( )
  :
}
```

isiliye hum left recursion eliminate karte hai

$\therefore \quad A \to A\alpha / \beta$

After eliminating left recursion we get

$$A \to \beta A'$$
$$A' \to \alpha A' \mid \varepsilon$$

Example: 1) $E \to E+T/T$

2) $T \to T*F/F$

1) $\underset{A}{E} \to \underset{A}{E}\underbrace{+T}_{\alpha}/\underbrace{T}_{\beta}$

jaisa app dekh sakte ho it contain left recursion

After eliminating left recursion we get

$$E \Rightarrow TE'$$
$$E' \to +TE'/\varepsilon$$

2) $T \to T*F/F$ is homework for you.

Multiple left recursion

$A \to A\alpha_1 | A\alpha_2 | A\alpha_3 | \beta_1 | \beta_2 | \beta_3$

After eliminating left recursion we gets

$A \to \beta_1 A' | \beta_2 A' | \beta_3 A'$

$A' \to \alpha_1 A' | \alpha_2 A' | \alpha_3 A' | \varepsilon$

Example:

$A \to Aa | Ab | Ac | e | f | g$

After eliminating left recursion we gets

$A \to eA' | fA' | gA'$

$A' \to aA' | bA' | cA' | \varepsilon$

Elimination of left factoring

$A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \mid \alpha\beta_3$

↓

After eliminating lefts factoring

↓

$A \rightarrow \alpha A'$

$A' \rightarrow \beta_1 \mid \beta_2 \mid \beta_3$

~~A → Sa|Sb|Sc~~

Example:

$A \rightarrow Sa \mid Sb \mid Sc$

After eliminating left factoring

$A \rightarrow SA'$

$A' \rightarrow a \mid b \mid c$