Date

Page

NAME:- KRUNAL RANK

ROLL NO:- U18CO081

CLASS:- BTECH III, Computer Eng.

SEM: Semester 6

U18C0081
21/01/2021

System Softwares

Tutorial 2

Ans 1: An interface is shared across boundary across which two or more components of a computer system exchange information. The exchange can be between hardware, softwares, peripheral devices, users or a combination of these.

Some types of interfaces are:-

- Hardware interfaces: These are characterized by mechanical, electrical and logical signals at the interface and protocol for sequencing them such as the buses.

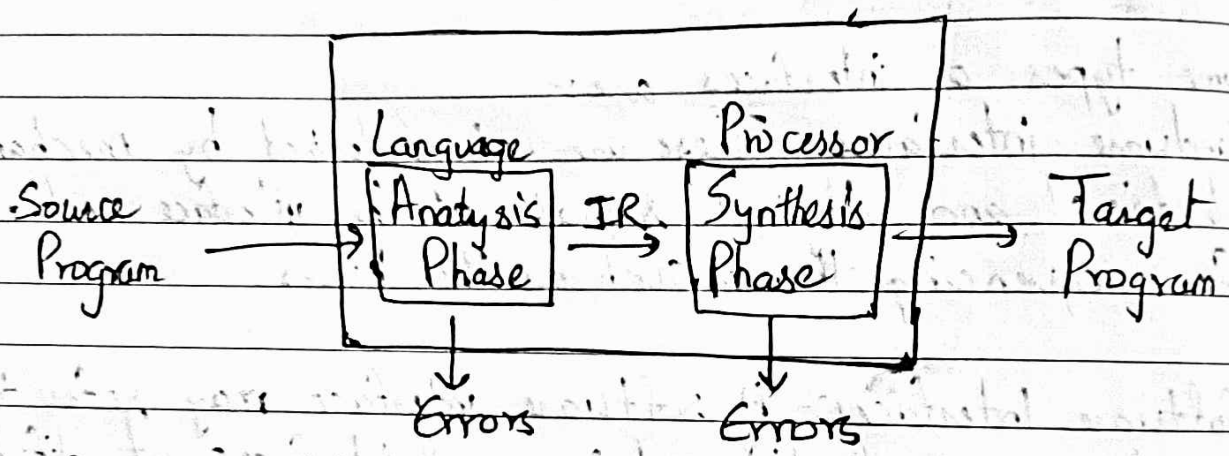
- Software Interfaces:- A software interface may refer to a vide range of different types of interfaces at different "levels". An operating system is an interface between the ~~hardware level~~ different types of hardware.

- User Interface:- A user interface is a point of interaction between computers and humans such as the GUI (Graphical User Interface) and CLI (Command Line Interface.

Ans 2: Language Processing = Analysis of Source Program
+
Synthesis of target program.

Analysis of Source Program:
It contains three types:-

1. Lexical Analysis: It governs lexical rules in the source language.
2. Syntax Analysis: It checks validity of the statements
3. Semantic Analysis: It checks the semantic rules in source language.



Source Program can be passed through analysis phase and generates Intermediate Representation which is further passed to synthesis phase which finally generates target program.

In analysis phase, we use ~~for~~ forwarding reference, which is a reference to the entity which precedes its definition in the program IR is having following properties:

1. Ease of use
2. Processing and memory efficiency.

Ans 3.

Compiler	Interpreter
· Compiler converts entire code to assembly language at a time.	· Interpreter converts line by line.
· It produces optimized code.	· Code is not as optimized as its compiled version.
· Program execution is faster.	· Program execution is relatively slow.
· Program analysis time is more.	· Program analysis time is less.
· Machine code is generated.	· No machine code is generated.
· Execution ~~gp~~ gap is present.	· No execution gap is present.

Ans 4.

· An intermediate representation is the data structure of a code used internally by a compiler or virtual machine to represent source code. It is designed to be conductive for further processing ~~so~~ such as optimization and translation.

Specification gap is the gap between application domain and programming language domain.

Execution gap is the gap between programming language domain and execution domain.

Ans 5: The life cycle of souce program defines the program behaviour and extends through execution stage which exhibits the behaviour specified in the program.

- Edit time: Phase where editing of souce program takes place, also known as designing phase.

- Compile time: The code after editing is passed on to a translater that translates it to ~~a~~ machine code.

- Distribution time: A program goes through ~~installation~~ distribution by an entity that created it to the entities that will execute it.

- Installation time: A program goes through installation phase, getting ready to be executed.

- Link time: The specific implementation of the interface is linked and associated to the program invoking it. System libraries are linked by using the lookup of the name and interface of the library needed.

- Load time: This stage actively takes the executible image from its stored repositories and places them into main memory for execution.
- Runtime: The final stage of life cycle of program in which its behaviour is demonstrated.