Voice Input

Voice input involves the computer "listening" for spoken words from the user.

1. This technology may be difficult to use due to the complexity of natural language
2. The computer may not understand all ways to phrase a single command.
3. The computer may not be able to understand different accents.
4. Background noise may render the user's voice unintelligible.

Types of Computer Interface

Voice Input

Touch Screen

Force Feedback

Virtual Reality

Augmented Reality

Assembly

Opcodes and Data

Opcode refers to the operative component of the machine code instruction. This might be **ADD**, where the data in this case would be the number to add(and would come after the opcode).

Mnemonics

This is a three-letter name given to an opcode in order to make assembly more human-readable.

Assembly (cont)

Translation to Machine Code

Assembly code is one-to-one with the mnemonic codes, but due to the existence of variables and labels translation becomes more difficult.

1. Labels are converted to memory addresses and stored in a symbol table(**not** the same kind as in **lexical analysis**, though!).
2. After that, mnemonics are converted into their binary counterparts and are combined with binary representations of data.

IDEs

Syntax Highlighting

The IDE may automatically highlight key words like "if" or "print", based on the language being used. Also, most IDEs will implement **automatic line numbering**.

Error Reports

If a syntax error has been detected, the IDE may attempt to point to the place in the code where the error has occurred.

Breakpoints

Breakpoints will make the program halt at a specific line of code.

Stepping Through

This involves making the program run through one line of code at a time and only proceeding when the developer presses a button to move to the next line.

Version Control

This allows potentially problematic or otherwise unwanted changes to be rolled back, or to create stable builds at a specific point in time and still allow development to continue.

System Design Evaluation

Systems Design Evaluation helps the company and client to determine whetherr the project was a success. This involves considering many criteria:

Requirements

Determine whether the system fits the objective specification, as established by the client at the beginning of the project.

Performance

Whether the program runs efficiently and effectively.

Robustness

This is whether the program crashes frequently or is generally stable.

Cost

Determine whether the system has cost more or less than expected.

Usability

Whether the end user is able to easily manoeuvre or use the program. This may depend on the user's level of expertise with computer software.

Link Editors and Link Loaders

What's a link editor?

This ensures that previously compiled libraries are loaded into a program at compile-time and the required links are created. The code from the libraries then becomes part of the final executable code.

What's a linking loader?

Rather than making the libraries specifically part of the final executable, a linking loader just loads the libraries when the program is executed or loaded into memory.

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Translators

A **Translator** is any piece of software which converts higher-level source code into machine code. These are **compilers**, **interpreters** or **assemblers**.

Compilers

1. Compilers convert source code from a high-level language into machine code. This generates an executable file which can be distributed.
2. This protects the intellectual property of the developers, since the source code is usually very difficult to reverse-engineer.
3. However, code compiled for one target platform(i.e, Windows) will not run on another(i.e Linux).

Interpreters

1. An interpreters translates source code into machine code **line-by-line**, executing each line as it goes.
2. This means that source code only has to be written once to run on any machine that has an interpreter for it.
3. It's easier to test single lines of code
4. Since the source code is compiled at run-time, this may slow down the system.
5. It's much easier for other people to view the source code, which may compromise intellectual property.

Assemblers

1. An assembler directly translates assembly mnemonics that each represents an instruction in machine code.

CASE Software Tools

CASE stands for **C**omputer-**A**ided **S**oftware **E**ngineering, and refers to a type of software which helps developers to create systems and programs. It may include any of the following:

Data Dictionary

A database of all data types, sizes and names used in the program.

Graphics/Diagram Production

Helps the developer to create **data flow diagrams** or **flowcharts**.

Libraries

Extra repositories of usable code, such as **SDL**.

Project Management Tools

This might include scheduling, task assignment, etc.

Version Control

Allows multiple developers to easily work together and keep track of historical changes to the program, like **Git**

Prototyping Tool

This helps the developer to quickly create screen designs, menus, etc without having to explicitly program them.

Augmented Reality

Augmented reality involves capturing the real world and overlaying information and graphics over it.

Force Feedback

This is also known as **haptic feedback**, and it refers to **vibrations** or other sensations that the computer may respond with once the user has performed an action(i.e, pressing a key on a touchscreen keyboard)

System Design Validation

Systems Design Validation is the process of testing a system to ensure that the designed product **meets the requirements**.

1. This is generally carried out as early in the project as possible.
2. It often involves things like bringing in customers to get feedback on the user interface.
3. Methods such as **Agile** generally make use of multiple smaller design reviews than a single large one.

CASE Tools

Upper CASE	These are tools which are useful during the **planning**, **analysis** or **design** sections.
Lower CASE	These are tools which are useful during the **implementation**, **testing** or **maintenance** sections.
Integrated CASE	These are tools which are useful in **all sections** of development.

Analysis and Planning Tools

Diagram Generator

Allows the developer to easily create diagrams and flowcharts which represent the flow of data within the program.

Requirement Recording

The developer can record and describe in detail any objectives or requirements for the system.

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Analysis and Planning Tools (cont)

Collaboration Manager

Allows multiple developers to easily work on the same program at the same time without causing merge conflicts. This is included in most **Lower CASE** tools.

Wire Framing

Allows the developer to easily prototype system logic, code structure and user interface.

Bug Tracking

Allows bug reports to be filed, assigned and eventually marked as "solved".

Virtual Reality

This involves "taking over" the user's senses in order to make them feel as though they have entered an alternate(or virtual) reality.

1. This is usually done via goggles, controllers and earpieces.
2. Each eye piece in the goggles relays a slightly different image to give the impression of a 3D image
3. However, this may cause some users to feel nauseous

Intermediate Code

Intermediate code is code that is compiled but not read directly by the end user's CPU. It's **interpreted** by an intermediate virtual machine which can then interact with the CPU.

1. This is generally faster than interpreted languages, since the instructions aren't being rapidly compiled to machine code but rather fed to an abstract/virtual CPU.
2. The code is cross-platform, as long as the same VM can be installed on many platforms.

Compilation

Lexical Analysis

1. Spaces, comments etc are removed.
2. All keywords, constants and identifiers are replaced with more space-efficient **tokens**. This is known as **tokenization**.
3. A symbol table is generated which holds the addresses of variables and functions.

Syntax Analysis

1. Determine whether the source code obeys the syntax rules of the language(i.e, variables can't start with numbers)

Semantic Analysis

1. Check for semantic errors, like ensure that all variables have been declared, types are used correctly(not multiplying strings), etc.

Machine Code Generation

1. This involves generating the actual machine code for use by the CPU.
2. Allocate the use of registers to minimize slow memory access.

Optimization of Machine Code

1. Optimize the code by limiting the number of jumps it has to do, "unrolling" loops, etc.

Touch Screens

Touch screens are screens that take in physical input from capacitive touch.

1. If selectable elements are too small or close together, it may be difficult to select with the average finger.
2. Touch screens are very intuitive, especially for small children.

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com