1. **Machine Language** - Also called low level language. All programs are converted to machine language before executed. It consists of 0 and 1 that represent high and low electrical voltage.
2. **Assembly Language** – A low level language. Consists of symbolic operaction code to represent machine code
3. **Operating System** – It is a set of commands that performs common system tasks like accepting data from keytboard, sending data to printer, etc. It is stored on hard disk and is loaded to memory when computer is turned on

**Assembler**

It is responsible for translating assembly language into machine code.

Functions

1. Translate mnemonic opcodes to machine language
2. Convert symbolic operands to their machine addresses
3. Error checking

**Loaders**

It is a program which takes object code as input and prepares it for execution and loads the executable code into memory.

**Linkers**

It combines two or more separate object programs and supplies the information needed to allow references between them.

**Explain types of text editor.**

* **Line editor**: In this, you can only edit one line at a time or an integral number of lines. Ex : Teleprinter, edlin, teco
* **Stream editors**: In this type of editors, the file is treated as continuous flow or sequence of characters instead of line numbers, which means here you can type paragraphs. Ex : [Sed editor](https://www.geeksforgeeks.org/sed-command-in-unix/) in UNIX
* **Screen editors**: In this type of editors, the user is able to see the cursor on the screen and can make a copy, cut, paste operation easily. It is very easy to use mouse pointer.   
  Ex : [vi](https://www.geeksforgeeks.org/vi-editor-unix/), emacs, Notepad
* **Word Processor**: Overcoming the limitations of screen editors, it allows one to use some format to insert images, files, videos, use font, size, style features. It majorly focuses on Natural language.
* **Structure Editor**: Structure editor focuses on programming languages. It provides features to write and edit source code. Ex :Netbeans IDE, gEdit.

**TWO PASS ASSEMBLER**

* Pass 1 scans the source program for label definitions and assigns addresses. In pass 1 location counter processing is done. Also symbols defined in the program are entered into symbol table.
* Pass 2 performs the actual translation into machine code. Second pass synthesizes the target using the information found in the symbol table.

**Two pass assembler performs following tasks**

Pass-1

1. Separate the symbol, mnemonic opcode and operand fields used
2. Construct symbol table
3. Construct intermediate representation

Pass-1

1. It processes the intermediate representation to synthesize the target program

**DATA STRUCTURES USED FOR PASS-1**

1. OPTAB – it consists of mnemonic opcodes and related information.
2. SYMTAB – it is symbol table. Used to store values assigned to labels
3. LITTAB – it is a table of literals

**Intermediate Code**

It consists of a set of IC units. Each unit has three fields:- Address, Opcode and Operands

**LOGICAL CLOCKS**

A logical clock of a process is a software counter that is used to timestamp events executed by the process so that the happened-before relation is respected by the timestamps.

Due to the absence of a [Global Clock](https://www.geeksforgeeks.org/logical-clock-in-distributed-system/) in a [Distributed Operating System](https://www.geeksforgeeks.org/types-of-operating-systems/) Lamport [Logical Clock](https://www.geeksforgeeks.org/logical-clock-in-distributed-system/) is needed

A person with one watch knows what time it is but a person with two or more watches is never sure. Lamport defined a relation happesn before(🡪).

**Conditions for happens-before:**

1. If a and b are events is the same process and a comes before b, then a 🡪b
2. If a is sending event of a message msg by one process, and b is the recipient event of msg, then a🡪b.
3. If a🡪b, b🡪c the a🡪c

**What is socket programming?**

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server.

1. **Compare compiler vs interpreter.**

|  |  |
| --- | --- |
| Interpreter translates just one statement of the program at a time into machine code. | Compiler scans the entire program and translates the whole of it into machine code at once. |
| An interpreter takes very less time to analyze the source code. However, the overall time to execute the process is much slower. | A compiler takes a lot of time to analyze the source code. However, the overall time taken to execute the process is much faster. |
| Interpreters are used by programming languages like Ruby and Python for example. | Compliers are used by programming languages like C and C++ for example. |

1. Explain macro vs subroutine.

|  |  |
| --- | --- |
| Macro can be called only in the program it is defined. | Subroutine can be called from other programs also. |
| Macro can have maximum 9 parameters. | Can have any number of parameters. |
| Macro can be called only after its definition. | This is not true for Subroutine. |
| A macro is defined inside: DEFINE … …. END-OF-DEFINITION. | Subroutine is defined inside: FORM ….. ….. ENDFORM. |
| Macro is used when same thing is to be done in a program a number of times. | Subroutine is used for modularization. |

**TWO PASS MACRO-PROCESSOR**

A macro processor is a program that reads a file (or files) and scans them for certain keywords. When a keyword is found, it is replaced by some text. The keyword/text combination is called a macro.

Input to the two pass macro processor is an assembly program with macro definitions and call and translate it into an assembly language which does not contain any macro definitions and calls

**DATA STRUCTURES**

1. Macro Name Table(MNT) – Holds the names of macro definitions in program. At the time of processing source program, the preprocessor compares the string found in mnemonic field to macro names in MNT
2. Actual Parameter Table(APT) – hold values of formal parameters. APT is split into

* Parameter Name Table(PNTAB) – contains formal parameters names. Parameters are entered in PNTAb in same order in which they appear.
* APTAB – contains formal parameters values.

1. Parameter Default Table(PDT) – holds name and default values of formal parameters.

**PHASES OF COMPILERS**

1. Lexical Analysis – also called scanning. The statement is broken up into tokens (like the assignment symbol, plus sign, constant no). Blank char are eliminated
2. Syntax Analysis – also called parsing. Tokens generated in lexical phase are grouped together in a hierarchichal structure called parse tree or syntax tree.
3. Semantic Analysis – once syntax is checked, the semantic analysis determines the meaning of the source code. Matching of parentheisis, if else statements etc

**STATIC AND DYNAMIC LINKING**

Static Link Libraries

* It is a routines or external func and variables which are solved by calling program at compile time
* For preparing an object file, static libraries are merged and from this a single executable file is created.
* Size of exe is greater than dynamic linking

Dynamic Link Libraries

* It is a collection of small programs any of which can be called by a large program running on computer
* They are not directly executable files
* They are separate files containing functions that are called by programs to perform certain job.
* Have extension .dll

**THREADS**

Thread is a basic processing unit to which an OS allocates processor time.

Every program has atleast one thread.

**Thread life cycle**

1. New
2. Ready
3. Running
4. Blocked
5. Waiting
6. Sleeping
7. Dead

**CPU SCHEDULING ALGO**

**Preemptive** – allows process to be interrupted in the midst of its execution, taking the CPU away and allocating it to different process.

**Non-Preemptive** – it ensures that the process gives its control over the CPU when it is completed.

**First Come First Serve Scheduling (FCFS)**

* The first process to request the processor gets it until finished.
* There is a single queue for ready processes.
* **It is Non-preemptive**
* Not useful in interactive processes
* Ex-buying tickets

**Shortest Job First Scheduling (SJFS) of Shortest Job Next (SJN)**

* It handles the process based on the length of their CPU cycle time.
* Reduces average waiting time of the process.
* Non-preemptive
* When a process request a CPU, it must inform the system how much time it wants. When CPU is available, the system allocates the processes with the least expected execution time.

**Priority Scheduling**

* CPU selects higher priority process first. If priority of two processes is same, it uses FCFS.
* Each process is assigned a priority number(non negative no).
* Both preemptive and non-preemptive

**Round Robin Scheduling**

* Preemptive algo
* Processes are given limited amount of time of the processor called time slice or time quantum. If process foes not complete in given time slixe, system gives CPU to next process and this process is send at the end of queue.

**REMOTE PROCEDURE CALL (RPC)**

* Developed by sun Microsystems
* It allows programs to execute subroutines on a remote system. The caller program which is the server sends a call message to the server process and waits for the reply.
* Main goal is to hide the existence of the network from a program.
* Well suited for client-server interaction in which the flow of control alternates between caller and callee

**Explain Deadlock.?**

Deadlock is a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process.

Consider an example when two trains are coming toward each other on the same track and there is only one track, none of the trains can move once they are in front of each other. A similar situation occurs in operating systems when there are two or more processes that hold some resources and wait for resources held by other(s). For example, in the below diagram, Process 1 is holding Resource 1 and waiting for resource 2 which is acquired by process 2, and process 2 is waiting for resource 1.

**What is mutual exclusion?**

Mutual exclusion is a property of process synchronization which states that “no two processes can exist in the critical section at any given point of time”.

**What is Concurrency?**

Concurrency is the execution of the multiple instruction sequences at the same time. It happens in the operating system when there are several process threads running in parallel

**What are the functions of operating system.**

Security – ...

Control over system performance – ...

Job accounting – ...

Error detecting aids – ...

Coordination between other software and users – ...

Memory Management – ...

Processor Management – ...

Device Management

**Explain process in detail.**

=>A process is an instance of a program running in a computer. It is close in meaning to task , a term used in some operating systems. ... A process can initiate a subprocess, which is a called a child process (and the initiating process is sometimes referred to as its parent ).

**What is scheduler**

==> scheduler is a software product that allows an enterprise to schedule and track computer batch tasks. ... Job schedulers may also manage the job queue for a computer cluster. A scheduler is one of the main components of IT infrastructure. A scheduler may also be known as a job scheduler

**What is virtual memory.?**

Virtual Memory is a storage allocation scheme in which secondary memory can be addressed as though it were part of the main memory. The addresses a program may use to reference memory are distinguished from the addresses the memory system uses to identify physical storage sites, and program-generated addresses are translated automatically to the corresponding machine addresses.

The size of virtual storage is limited by the addressing scheme of the computer system and the amount of secondary memory is available not by the actual number of the main storage locations.

**What is paging ?**

Paging is a memory management scheme that eliminates the need for contiguous allocation of physical memory. This scheme permits the physical address space of a process to be non – contiguous.

**Explain demand pagging.?**

Demand paging is a process of swapping in the Virtual Memory system. In this process, all data is not moved from hard drive to main memory because while using this demand paging, when some programs are getting demand then data will be transferred. But, if required data is already existed into memory then not need to copy of data. The demand paging system is done with swapping from auxiliary storage to primary memory, so it is known as “Lazy Evaluation”.

**What is thrashing.?**

Thrashing is a computer activity that makes little or no progress.Usually, this happens either of limited resources or exhaustion of memory.

It arises when a page fault occurs.Page fault arises when memory access of virtual memory space does not map to the content of RAM.

The effect of Thrashing is

1.CPU becomes idle.

2.Decreasing the utilization increases the degree of multiprogramming and hence bringing more processes at a time which in fact increases the thrashing exponentially