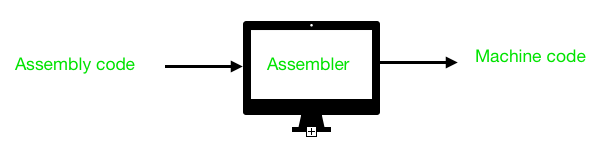
**Lab Practice I Practical Question Bank**

1. **What is assembler? Use of assembler?**

Assembler is a program for converting instructions written in low-level assembly code into relocatable machine code and generating along information for the loader.



* **Pass-1:**
  1. Define symbols and literals and remember them in symbol table and literal table respectively.
  2. Keep track of location counter
  3. Process pseudo-operations
* **Pass-2:**
  1. Generate object code by converting symbolic op-code into respective numeric op-code
  2. Generate data for literals and look for values of symbols

1. **Pass-I Assembler Data structures and their use?**

OPTAB,SYMTAB,LITTAB,POOLTAB

1. **What is the input for pass-I assembler? And pass-II Assembler?**

Src code is the input to p1a and the intermediate code which is generated by p1a is the input to the p2a

1. **What is macro? Need for Macro? Advantages of macro?**

A [**macro**](https://www.geeksforgeeks.org/c-language-2-gq/macro-preprocessor-gq/) is a piece of code in a program that is replaced by the value of the macro.

Advantages : It reduces the length of the program.

The speed of the execution of the program is the major advantage of using a macro.

1. **Data structures used in Macro Pass-I?**

Deftab,namtab,argtab

1. **What do you mean by loader linker?**

Linker : A linker is special program that combines the object files, generated by compiler/assembler and other pieces of code to originate an executable file has .exe extension

Loader : It is special program that takes input of executable files from linker, loads it to main memory, and prepares this code for execution by computer. Loader allocates memory space to program.

1. **What is the use of compiler?**

compiler, computer software that translates (compiles) source code written in a high-level language (e.g., C++) into a set of machine-language instructions that can be understood by a digital computer's CPU. Compilers are very large programs, with error-checking and other abilities.

1. **Difference between Compiler and Interpreter?**

| S.No. | Compiler | Interpreter |
| --- | --- | --- |
| 1. | The compiler scans the whole program in one go. | Translates the program one statement at a time. |
| 2. | As it scans the code in one go, the errors (if any) are shown at the end together. | Considering it scans code one line at a time, errors are shown line by line. |
| 3. | The main advantage of compilers is its execution time. | Due to interpreters being slow in executing the object code, it is preferred less. |
| 4. | It converts the source code into object code. | It does not convert source code into object code instead it scans it line by line |
| 5 | It does not require source code for later execution. | It requires source code for later execution. |

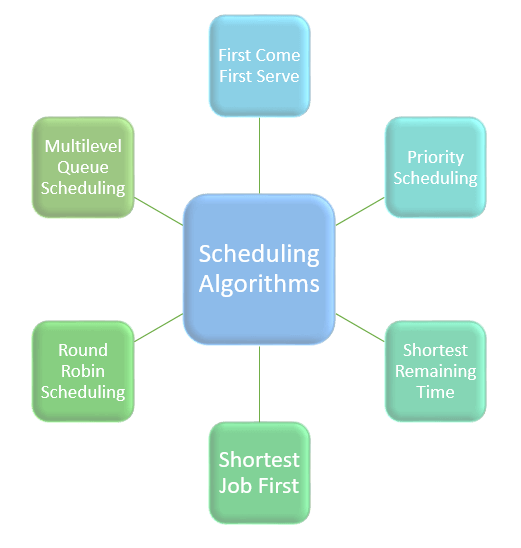
1. **Difference between macro and subroutine?**

|  |  |
| --- | --- |
| **Macro** | **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. |

1. **What do you mean by CPU Scheduling.**

CPU scheduling is the task performed by the CPU that decides the way and order in which processes should be executed. There are two types of CPU scheduling - Preemptive, and non-preemptive.

1. **How many algorithms are used in CPU Scheduling? Explain each with an example.**



1. **Difference between Preemptive and non-preemptive.**

| Parameter | PREEMPTIVE SCHEDULING | NON-PREEMPTIVE SCHEDULING |
| --- | --- | --- |
| Basic | In this resources(CPU Cycle) are allocated to a process for a limited time. | Once resources(CPU Cycle) are allocated to a process, the process holds it till it completes its burst time or switches to waiting state. |
| Interrupt | Process can be interrupted in between. | Process can not be interrupted until it terminates itself or its time is up. |
| Overhead | It has overheads of scheduling the processes. | It does not have overheads. |
| Flexibility | flexible | rigid |

1. **Which CPU scheduling algorithm is best and why?**

Round Robin is Best because

A big advantage of round robin scheduling over non-preemptive schedulers is that **it dramatically improves average response times**. By limiting each task to a certain amount of time, the operating system can ensure that it can cycle through all ready tasks, giving each one a chance to run.

1. **What is priority scheduling?**

Priority Scheduling is a method of scheduling processes that is based on priority. In this algorithm, the scheduler selects the tasks to work as per the priority.

The processes with higher priority should be carried out first, whereas jobs with equal priorities are carried out on a round-robin or FCFS basis.

1. **What do you mean by Memory placement strategies?**

Determine where in main memory the system should place incoming program or data pieces, first fit, best fit, and worst fit.

1. **List out all strategies.**

**First Fit** : In the first fit, the partition is allocated which is first sufficient from the top of Main Memory.

**Next fit** is a modified version of [‘first fit’](https://www.geeksforgeeks.org/program-first-fit-algorithm-memory-management/). It begins as the first fit to find a free partition but when called next time it starts searching from where it left off, not from the beginning. This policy makes use of a roving pointer. The pointer moves along the memory chain to search for a next fit.

**Best fit** allocates the process to a partition which is the smallest sufficient partition among the free available partitions.

**Worst Fit** allocates a process to the partition which is largest sufficient among the freely available partitions available in the main memory. If a large process comes at a later stage, then memory will not have space to accommodate it.

1. **Which one is best?**

Best Fit

1. **Difference between first and best fit?**

Refer 16

1. **Explain Worst fit and next fit?**

Refer 16

1. **Explain sensors used in practical**
2. **Real time applications of IOT**
3. **What is Gant chart? Explain its use?**
4. **What are different reporting tools? Which one is popular?**
5. **What is GOMS? Explain in detail.**