

AS-2883
B.Sc.(Hon's)(Fifth Semester) Examination,2013
Computer Science
(PCSC-503)
(System Software)

[Time Allowed: Three Hours]

[Maximum Marks : 30]

Note: Question Number 1 is compulsory.

Marks : 10X1

Q1.Following questions are having four options. Choose the correct option.

(i) Which of the following is not register :

- A. Accumulator
- B. Status word
- C. Program counter

D. Buffer

(ii) In two pass assembler the pseudo-code EQU is to be evaluated during :

A. Pass 1

- B. Pass 2
- C. Not evaluated by the assembler
- D. None of above

(iii) Pick the correct statement about LTORG

- A. It is used to load the object program at some specified memory location that is given in the operand field
- B. It is used to place the literals definition in a specified memory location that is given in the operand field

C. It creates a literal pool that contains all the literal operands used since the previous LTORG

D. None of above

(iv) In a compiler grouping of character into tokens is done by the

A. Scanner

- B. Parser
- C. Code generator
- D. Code optimiser

(v) An analysis, which determines the syntactic structure of the source statement, is called

- A. Semantic analysis
- B. Process analysis

C. Syntax analysis

D. Function analysis

(vi) A linker program

- A. Places the program in the memory for the purpose of execution.
- B. Relocates the program to execute from the specific memory area allocated to it.
- C. Links the program with other programs needed for its execution.
- D. Interfaces the program with the entities generating its input data.

(vii) Nested Macro calls are expanded using the

- A. FIFO rule (First in first out)
- B. LIFO (Last in First out)
- C. FILO rule (First in last out)
- D. None of the above

(viii) Which of the following about the loader is incorrect?

- A. Loader brings object program into memory for execution
- B. Dynamic linking scheme delay linking until execution time
- C. Absolute loader modifies the object program so that it can be loaded at any address location
- D. None of above

Note - Answer the following question.

(IX) Write name of the two versions of SIC?

Ans: Two versions of SIC are

1. Standard model
2. XE version(Extra Equipment) or (Extra Expensive)

(X) Write function of assembler directive RESB?

Ans: Reserve the indicated number of bytes for a data area

NOTE- Attempt Any FOUR

Marks: 4X5

Q2. Write a program to compute $A+B-1$ and store the result in 'C' for SIC.

Ans:

Source statement			Function
	LDA	A	Load value of 'A' into Accumulator
	ADD	B	Add value of B with the accumulator content
	SUB	ONE	Subtract one from the accumulator content
	STA	C	Store the value of " $A+B-1$ " in "C"
A	RESW	1	Qne word variable
B	RESW	1	Qne word variable
C	RESW	1	Qne word variable
ONE	WORD	1	One word constant

Q3. What is addressing mode? Explain following addressing modes

(a) Immediate addressing mode

(b) Direct addressing mode

(c) Indirect addressing mode

Ans:

Addressing Modes- The way the operand are chosen during program execution is depend on the addressing mode of the instruction. The addressing mode specifies a rule for interpreting or modifying the address field of the instruction before the operand is actually referenced. Computers use addressing mode technique for the purpose of accommodating one or both of the following provision:

(1) To give programming versatility to the user by providing such facilities as pointers to memory, counter for loop control, indexing of data, and program reallocation.

(2) To reduce the number of bits in addressing field of the instruction.

Immediate addressing mode:-In this addressing mode the operand is specified in the instruction itself. In other word, an immediate –mode instruction has an operand field rather than an addressing field

Target address=operand value

Direct addressing mode:-In this mode the effective address is equal to the address part of the instruction. the operand resides in the memory and its address is given directly by the address field of the instruction.

Target address=location of operand value

Indirect addressing mode:-In this mode the address field of the instruction field of the instruction gives the address where the target address is stored in the memory.

Target address=location of the location of operand value

Q4. Write algorithm for Pass 1 and Pass 2 of ASSEMBLER.

Ans:

Pass 1

1. Read input line.
2. Check to see if the opcode in the line is start.
 - I. Find if there is any operand field after start, then that indicates the location from which the program has to be loaded in the memory ,so initialize the location counter to the operand value.
 - II. Otherwise the location counter is set zero and the program starts loading from the starting address zero.
3. Write the line to the intermediate line.
4. Now start following processing for the other lines in the program , if is not comment line , until the op-code is "END".

IF there is a symbol or label in the label field.

- I. Check the symbol table to see if the symbol has already been stored over there. If so, a duplicate symbol error message should be displayed.
- II. Otherwise the symbol is entered in to the Symbol table, along with memory address in which it is stored.

If there is an opcode in the line read.

- I. Search the Operation table to see if the opcode is present, if not the opcode written is not of valid syntax, so error message should be displayed.
 - II. If the opcode is present in Operation table add instruction length to location counter.
 - III. If the opcode is Byte , WORD , RESW ,RESB add operand length to location counter.
5. Write each and every line processed to the intermediate file.
 6. Subtract the starting address of the program from the final value of the location counter , to calculate the length of the program.

Pass 2

1. Read the first line from the intermediate file.
2. Check to see if the opcode from the first line read is start. IF so then write the header record to the object program.
3. Start the following processing for the other lines in the intermediate file it is not a comment line until an "END" statement is reached.
 - I. Start writing the text record to the output file. If symbol is present in the operand field of the instruction, assemble opcode of the instruction from the Operation table ,with the address of the symbol from the symbol table.
 - II. If the symbol is present in the operand field, and if does not occupy any place in the symbol table then undefined error message should be displayed.
 - III. If there is no symbol in the operand field ,then the operand address is assigned as zero, and it is assembled with the object code of the instruction.
 - IV. If the opcode is BYTE, WORD, RESB etc convert the constants to the object code
4. When all the text records have been written to the assembly listing write the END record.

Q5. What are the applications of Dynamic Linking? Explain the process involved in Dynamic Loading and Linking routines with suitable diagram.

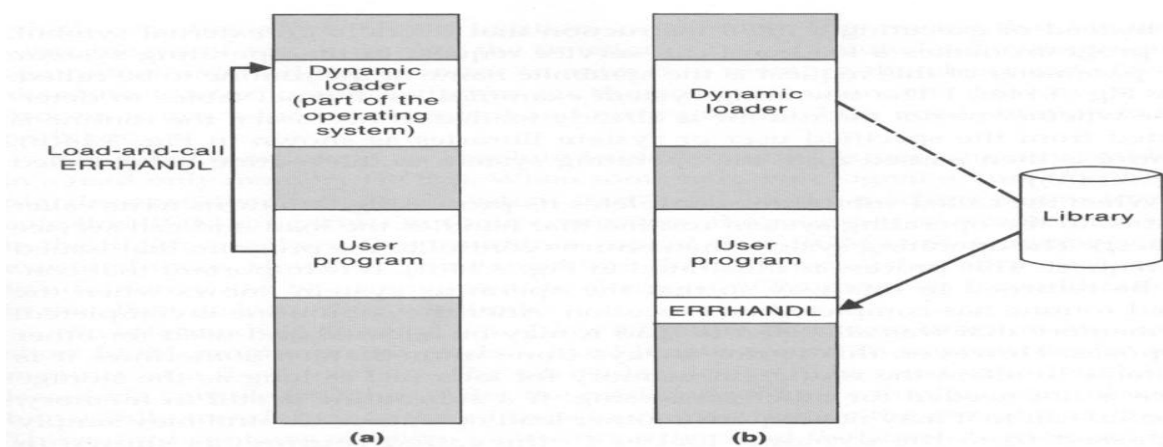
Applications of Dynamic Linking-

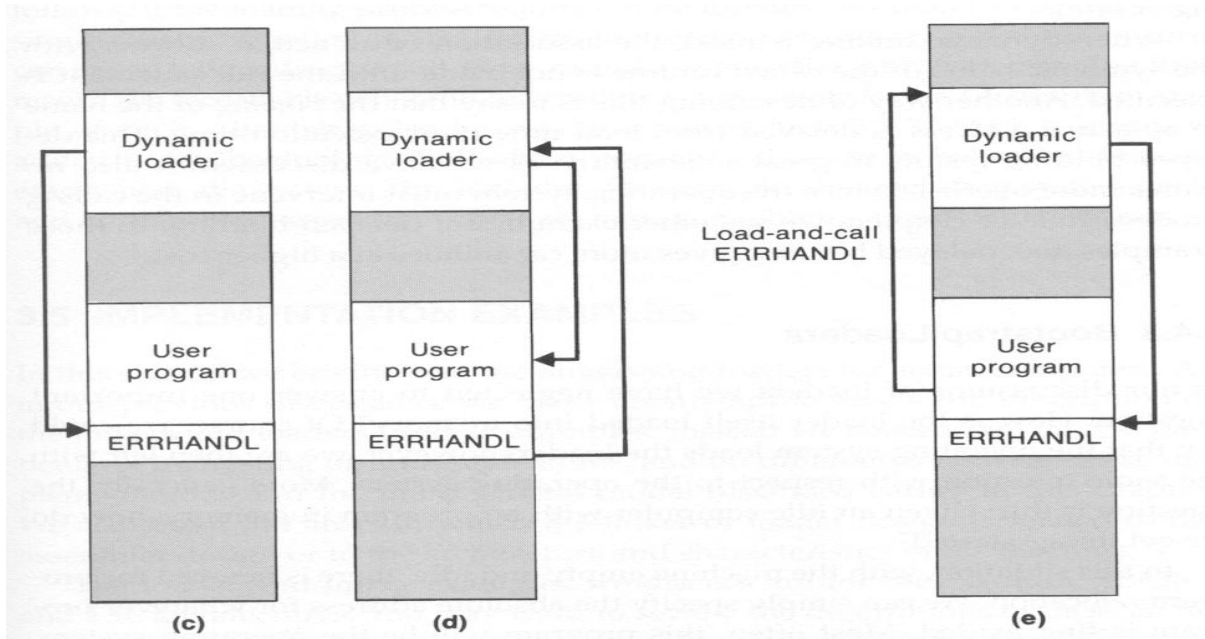
- Dynamic linking postpones the linking function until execution time.
 - A subroutine is loaded and linked to the test of the program when it is first called.
- Dynamic linking is often used to allow several executing programs to share one copy of a subroutine or library.
- For example, a single copy of the standard C library can be loaded into memory.
- All C programs currently in execution can be linked to this one copy, instead of linking a separate copy into each object program.

- In an object-oriented system, dynamic linking is often used for references to software object.
- This allows the implementation of the object and its method to be determined at the time the program is run. (e.g., C++)
- The implementation can be changed at any time, without affecting the program that makes use of the object.
- Using dynamic linking can save both space for storing the object program on disk and in memory, and time for loading the bigger object program.

Process involved in Dynamic Loading and Linking-

- A subroutine that is to be dynamically loaded must be called via an operating system service request.
 - This method can also be thought of as a request to a part of the loader that is kept in memory during execution of the program
- Instead of executing a JSUB instruction to an external symbol, the program makes a load-and-call service request to the OS.
- The parameter of this request is the symbolic name of the routine to be called.
- The OS examines its internal tables to determine whether the subroutine is already loaded.
- If needed, the subroutine is loaded from the library.
- Then control is passed from the OS to the subroutine being called.
- When the called subroutine completes its processing, it returns to its caller (operating system).
- The OS then returns control to the program that issues the request.
- After the subroutine is completed, the memory that was allocated to it may be released.
- However, often this is not done immediately. If the subroutine is retained in memory, it can be used by later calls to the same subroutine without loading the same subroutine multiple times.
- Control can simply pass from the dynamic loader to the called routine directly.





Q6. Explain recursive macro expansion with examples.

Recursive micro expansion:-

1. Invocation of one macro by another is said to be recursive macro expansion or nested macro calls.
2. Recursive macro expansion can easily be accomplished by writing the macro processor in programming language like pascal or c that allows recursive calls
3. We refer to the macro containing the nested calls as the outer macro and the called macro is inner macro.
4. Expansion of nested macro calls follow the last in first out (LIFO) rule.
5. Thus in the structure of nested macro calls, expansion of the latest macro call is completed first (the inner most macro call in the structure is completed first).

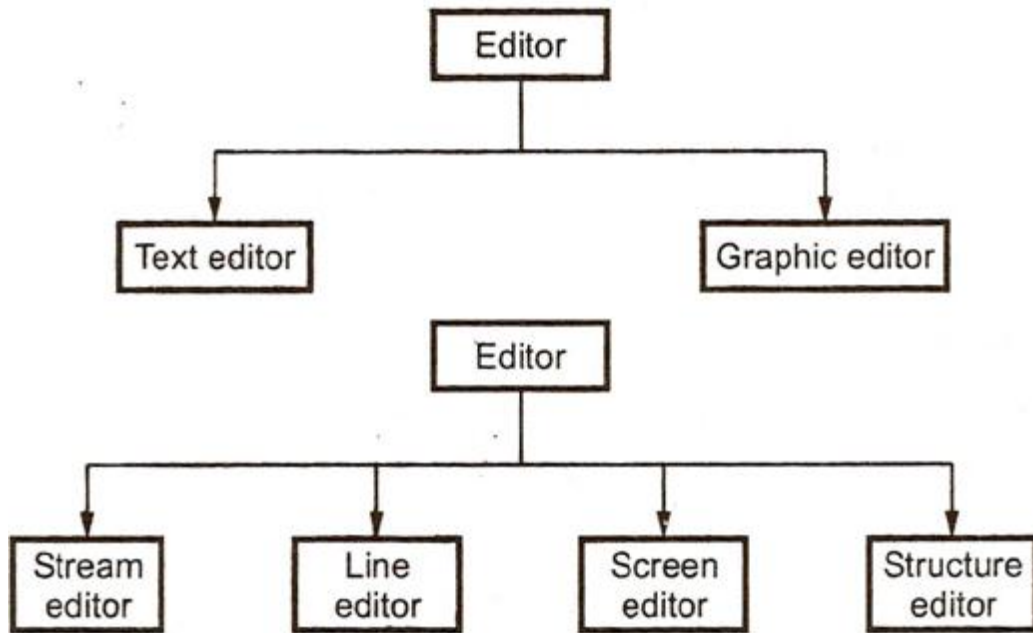
Example

Line	Source statement		
1	PROG	START	0
2	INPUT	MACRO	&A1,&A2
3		CLEAR	A
4		CLEAR	S
5		SUM	&A1,&A2
6		MEND	
7	SUM	MACRO	&B1,&B2
8		LDA	&B1
9		LDS	&B2
10		ADDR	S,A
11		MEND	
12		INPUT	05,06
13		END	PROG

Q7. Describe different types of editors in detail.

Editors: For any computing environment interactive text editors has an important part.

- Various types of editors are:



(Types of editors)

- Text editor should be the primary interface to the computer for all types of knowledge workers.

Overview of the editing process'

- An interactive editor is a computer program that allows a user to create and revise a target document.

- The document means an object such as computer programs, text, equations tables, diagrams, photographs etc.

- The document editing process in an interactive user computer dialogue designed to accomplish four tasks:

- 1) Select the part of the target document to be viewed & manipulated.
- 2) Determine how to format this view online and how to display it.
- 3) Specify and execute operations that modify the target document.
- 4) Update the view appropriately.

Screen Editors:

- Screen editor uses the what you see is what you get principle in editor design.

- A editor displays a screenful of text at a time. The user can move the cursor over the screen, position it at the point where user desires to perform some editing and proceed with the editing directly.

- The user has full control over the entire terminal. For example over an type exiting string which user wishes to replace. User can bring the cursor over a character to be deleted and press a delete key.

- It is possible to see the effect of an edit operation on the screen.

Line Editor:

- Line editor is one of the simplest type of editor which uses a buffer to store information:

- It operates in command mode. User give the command to the editor for any operation. Editor will respond this command.

- Buffer is in the main memory. That is set aside to store the information which is entered from the keyboard.

Merits

- 1) Simple for read and write.
- 2) Simple in design and implementation
- 3) Command and responses are interleaved.

Demerits

- 1) Not user friendly
- 2) Context of the text is not displayed.
- 3) Only single line is used for editing.

Stream Editor:

- A stream editor views the entire text as a stream of characters. This permits edit operations to cross line boundaries.
- Stream editor typically support character line and context orient commands.
- In stream editor, the current editing context indicated by the position of text pointer. This pointer can be manipulated using positioning.
- Stream editor maintain multiple representations.

Word Processors:

- It is document editors with additional features. It produce well formatted output.
- Features of text
 - a) Merging of text
 - b) Searching and replacement of word
 - c) Moving sections of text

Q8. Find Equivalent Object code for Statement in Line No. 2 using Program counter relative addressing mode (Use OPCODE for STL is 14)

Line No.	Location	Source Statement		
➔ 2	0000	FIRST	STL	RETADR
3	0003		LDB	#LENGTH
.....				
.....				
18	0030	RETADR	RESW	1

Ans:-

As Program Counter Relative Addressing is used in statement with line number '2',

n i x b p e

Opcode	1	1	0	0	1	0	Displacement
--------	---	---	---	---	---	---	--------------

displacement= RETADR - PC = 30-3 = 2D

STL machine language equivalent is 14.

First two bits are '1', Hexadecimal equivalent of 11 is 3.

14+3=17

Hexadecimal equivalent of '0010 0000' is 20.

Equivalent Object code=**17202D**