Arab Academy For Science and Technology & Maritime Transport

College of Engineering & Technology

Computer Engineering Department

EXAMINATION PAPER – Week 12

Course Title: System Programming

Course Code: CC410

Date: Thurs. Dec, 18-2014 Lecturer: Dr. Manal Helal

Time allowed: 60 Minutes Start Time: 12:30 p.m.

Student's name: Reg.# :

Question #	Marks	
	Available	Actual
Assemblers	12	
Loaders	8	
Total	**20**	

Lecturer	Name : Dr. Manal Helal
	Signature :
	Date:

MPC6/1-1

Assembler: **[12 points]**

1) What is the difference (if any) between the following A and B sequences of statements? [4 points]

(I)	(II)
A. VALUE EQU 10 ... LDA #VALUE	A. LDA LENGTH SUB 1
B. LDA #10	B. LDA LENGTH-1

2) Which problems should we solve if we want to design a one-pass assemblers? [4 points]

3) Program Block:

a) Please explain Program Block briefly. [1 points]

b) Which directive will be used in source program to identify the block? [1 point]

c) What kind of advantage will be gained by using Program Block? [2 points]

Loaders: [8 points]

4) Describe the absolute loader and how it works, and why is it useful. [3 points]

5) Program Relocation:

a) Define Program Relocation? [1 points]

b) How it affects loading the program? [1 points]

c) Describe two methods used in loading relocateable programs [3 points]

Answers:

1):

I. Instead of loading the literal 10 directly in Accumulator register A in B, a symbol is defined using EQU assembler directive and is loaded in the accumulator in A. Both (A & B) code fragments achieves the same goal.

II. If length is defined by address 0033 with the value 5. The result of the given statement in A, Accumulator register A is loaded with value 5 and subtracted by 1, thus giving the result 4. In B, Accumulator register A is loaded with value defined in location 0032. The goal from the code in A is different from the goal from the code in B.

2) The main problem is **forward referencing [2 marks]**. Either eliminate them or design an assembler that places in memory all unresolved symbols with links to where they are referenced, so that once the symbol is resolved, all references are updated with its address **[2 marks]**. More in the book, slides and notes.

3) a) Program Blocks allow the generated machine instructions and data to appear in the object program in a different order. Assembler rearranges these segments to gather together the pieces of each block and assign address (LOCCTR).

b) USE [blockname]

c) Separating the program into blocks in a particular order is useful such that program readability is better if data areas are placed in the source program close to the statements that reference them, and hence their addressing (LOCCTR) will be closer and less than 4096, requiring format 3 instruction format. This will make the object file smaller.

4) Absolute loader loads an object program in the specified address locations. It is useful for loading programs in simple machine such as SIC, in which no multiprogramming is allowed, no relocation is performed. It is also useful to load the operating system to specified location in memory for more sophisticated machines, such as the bootstrap loader.

5) a) program relocation allows the program to be loaded where the operating system assigns a free address for it. This feature allows multiprogramming capability, such that a machine can load and execute several programs in several blocks in memory.

b) The loader should read the ROGADDR, which is the beginning address in memory where the linked program is to be loaded. The OS supplies its value to the loader.

c) All addresses in the object program need to be modified by the starting address of the whole program. All direct addressing need to be modified by M records in case of machines such as SIC/XE, or relocation bit-mask in case of simple machines such as SIC. More in the book, slides and notes.