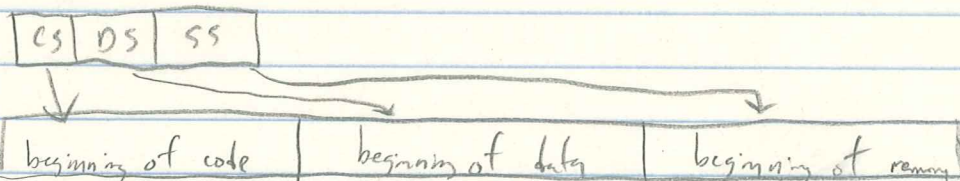


* Quiz 1 & 2 Review

- A in ASCII is 41h, 65d
- a in ASCII is 61h, 97d
- A in Hollerith is 12-1, EBCDIC is C1h
- J in Hollerith is 11-1, EBCDIC is D1h (not direct translation)
- C1-C9, D1-D9, E2-E9 - EBCDIC alphabet
- 1d in EBCDIC is F1h, ASCII is 31h
- 0d in ASCII is 30h
- Remember that a digit in a base can not be larger or equal to the base
- Long quiz on Thursday! 9.8.11

* Notes

- register - high speed memory that perform arithmetic, holds instructions
- warehouse registers - A, B, C, & D
- 8 bit - AH & AL
- 16 bit - AX, BX, CX, & DX
- 32 bit - EAX, EBX, ECX, EDX
- 64 bit - RAX, RBX, RCX, RDX
- CS, DS, SS - 16 bit
- CS - code, DS - data, S - segment (stacks)
- CS gives us a description



- index registers - SI & DI
- SI - source index, DI - destination index
- used to hold starting address of something

Assembly

9.6.11

Page 2

(SCI 2162)

- index registers are 16 bit
- ESI, EDI - extended index registers, 32 bit
- RSI, RDI - 64 bit index registers
- pointer registers (no 8 bit)
- BP - base pointer, keeps track of where stack was at time a piece of code ^{is} entered
- SP - stack pointer, points to top of stack
- IP - instruction pointer
- IP is not to be messed with, but hackers uses IP to inject code
- IP is updated automatically by operating systems
- all of these have extended registers - EBP, ESP, EIP
- .data, .stack, .code - directives
- .stack 100h - allocates stack of 256 bytes

- 1 byte db
- 2 word dw
- 4 dword dd
- 8 qword dq

- .data

				location counter
bVar	byte	?	(00	0000
wVal	word	27	001B	0001
dVal	dword	-7	FFFFFFF9	0003
qNum	qword	19	00000000 00000013	0007
				000F

- assembler creates symbol table
- symbol - location counter - size

Assembly

CSCI 2160

- if you define an array of values, the location counter reflects that 1 to 1
- the size table reflects the size of what is declared (byte, word, dword, etc.)

loc ctr	.data	equivalent hex	size
0000	byte 9, 7	09, 07	1
0002	word 17, -2, 3	0011, FFFE, 0003	2
0008	dword 29, 32	0000001D, 00000020	4
0010	qword 5	00000000 00000005	8
0018	byte '123A'	31 32 33 41	1
001C			

- in memory ...

.data

09 07

11 00 FE FF 03 00

1D 00 00 00 20 00 00 00

05 00 00 00 00 00 00 00

31 32 33 41

- dup directive (used with # in front)
- 2 dup (?) - 00 00 (byte)
- 3 dup (7) - 0007 0007 0007 (word)

Assembly

CSC 2160

- wX word 33, 19
 wY word 26, 39, 55
 wZ word ?
 wA word 2 dup (5)
 dVal dword 3 dup (-19)
 dStr byte 2 dup ('ABC')

loc chr	actual hex	reverse byte order
0000	0021, 0013	21 00 13 00
0004	001A, 0027, 0037	1A 00 27 00 37 00
000A	0000	00 00
000C	0005, 0005	05 00 05 00
0010	FFFFFFED, FFFFFFFE, FFFFFFFE	ED FF FF FF ED FF FF FF ED FF FF FF
001C	41 42 43 41 42 43	41 42 43 41 42 43
0022		

- Quiz on Thursday will cover everything up to 9.6.11