National University of Computer and Emerging Sciences

Lahore Campus

Computer organization and assembly language Date: November 5th 2024 Course Instructor(s) AA,AA,SF,SI,SM Moz 2AMHU Student Signature Instructions: Attempt all questions. It is an open book exam only Assembly Language Programming Lecture Notes are allowed. Calculators are allowed. You can use rough sheets. CIO #:2 Describe the working of important x86 assembly primitives, including arithmetic, branching, bit manipulation, addressing modes and interrupt handling. Office of ISR:0x000 (a) The segment and offset of interrupt service routine of int 77h are placed at: offset: MANIMAL STREET SERVICE (IN Which registers are changed by the iret instruction? Up Co Sp Mox ax (c) Which interrupt is hooked by the instructions given on right side? (e)Replace the following independent invalid instructions with a single instruction that has the same effect. (i) mov ax, (ss:sp) Solution: posh ox poon bx (f)Write a code to swap two registers ax and bx without using temporary space (local variable) on stack or memory. You are only allowed to use stack operations. (g)What would be the value of SP after Interval Marks: 40 Total Marks: 40 Total Marks: 40 Total Marks: 40 Total Questions: 2 Signature Student Signature Signature Total Questions: 2 Code for part (d) mov ax, 0 mov ax, 0 mov ex, ax mov [es:400], mySub mov [es:402], cs Mov [es:402]	COAL (FF2002) N	A Sessional-II Exam
Date: November 5th 2024 Course Instructor(s) AA,AA,SF,SI,SM Ploz 2APPHOL BCG - 3E Roll No Instructions: Attempt all questions. It is an open book exam only Assembly Language Programming Lecture Notes are allowed. Calculators are allowed. You can use rough sheets. CLO #:2 Describe the working of important x86 assembly primitives, including arithmetic, branching, bit manipulation, addressing modes and interrupt handling. Q1: [marks 4x2 + 2x2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2	The state of the s	10 %
Course Instructor(s) AA,AA,SF,SI,SM Moz 2AHHUL BCG-3E Student Signature Instructions: Attempt all questions. It is an open book exam only Assembly Language Programming Lecture Notes are allowed. Calculators are allowed. You can use rough sheets. CLO #:2 Describe the working of important x86 assembly primitives, including arithmetic, branching, bit manipulation, addressing modes and interrupt handling. Q1: [marks 4x2 + 2x + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2	Computer organization and assembly language	
AA,AA,SF,SI,SM Moz 2AM HILL 31 - 1001 Section Section Student Signature Instructions: Attempt all questions. It is an open book exam only Assembly Language Programming Lecture Notes are allowed. Calculators are allowed. You can use rough sheets. CLO #:2 Describe the working of important x86 assembly primitives, including arithmetic, branching, bit manipulation, addressing modes and interrupt handling. Q1: [marks 4x2 + 2x2 + 2 + 2 + 2 + 2 = 20] (a) The segment and offset of interrupt service routine of int 77h are placed at: offset: ANGULAN Segment: Oxford Code for part (d) (b) what is the total size of the IVT table? Oxford Code for part (d) (c) Which registers are changed by the iret instruction? Oxford Code for part (d) (d) Which interrupt is hooked by the instructions given on right side? (e) Replace the following independent invalid instructions with a single instruction that has the same effect. (i) mov ax, [ss:sp] add sp, 2 iii) sub sp, 2 mov [ss:sp], ax (f) Write a code to swap two registers ax and bx without using temporary space (local variable) on stack or memory. You are only allowed to use stack operations. Solution: Desh ax pop bx (g) What would be the value of SP after Jmp start		Total Walks
Roll No Section Student Signature Instructions: Attempt all questions. It is an open book exam only Assembly Language Programming Lecture Notes are allowed. Calculators are allowed. You can use rough sheets. CLO #:2 Describe the working of important x86 assembly primitives, including arithmetic, branching, bit manipulation, addressing modes and interrupt handling. Q1: [marks 4x2 + 2x2 + 2 + 2 + 2 + 2 + 2 = 20] (a) The segment and offset of interrupt service routine of int 77h are placed at: offset: ANOMAN Segment: ANOMAN OF SEGMENT OF SEGME		Total Questions: 2
Roll No Section Instructions: Attempt all questions. It is an open book exam only Assembly Language Programming Lecture Notes are allowed. Calculators are allowed. You can use rough sheets. CLO #:2 Describe the working of important x86 assembly primitives, including arithmetic, branching, bit manipulation, addressing modes and interrupt handling. Offset of Iss. Oxolog. Q1: [marks 4x2 + 2x2 + 2 + 2 + 2 + 2 = 20] 7	AA,AA,SF,SI,SM	
Roll No Section Student Signature Instructions: Attempt all questions. It is an open book exam only Assembly Language Programming Lecture Notes are allowed. Calculators are allowed. You can use rough sheets. CLO #:2 Describe the working of important x86 assembly primitives, including arithmetic, branching, bit manipulation, addressing modes and interrupt handling. Q1: [marks 4x2 + 2x2 + 2 + 2 + 2 + 2 + 2 = 20] /	102 2011	: \\\
Instructions: Attempt all questions. It is an open book exam only Assembly Language Programming Lecture Notes are allowed. Calculators are allowed. You can use rough sheets. CLO #:2 Describe the working of important x86 assembly primitives, including arithmetic, branching, bit manipulation, addressing modes and interrupt handling. Q1: [marks 4x2 + 2x2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 = 20] /	231 1000	Student Signature
Lecture Notes are allowed. Calculators are allowed. You can use rough sheets. CLO #:2 Describe the working of important x86 assembly primitives, including arithmetic, branching, bit manipulation, addressing modes and interrupt handling. Office of Isr. Oxoroc oxoro	Kullito	
CLO #:2 Describe the working of important x86 assembly primitives, including arithmetic, branching, bit manipulation, addressing modes and interrupt handling. Office of Isr. 3 xoloc to Isr. 4 xoloc to Isr.		
Q1: [marks 4x2 + 2x2 + 2 + 2 + 2 + 2 + 2 + 2 = 20] (a) The segment and offset of interrupt service routine of int 77h are placed at: offset: Axamamamamamamamamamamamamamamamamamamam		ssembly primitives, including arithmetic,
(a) The segment and offset of interrupt service routine of int 77h are placed at: offset:	branching hit manipulation, addressing modes a	and interrupt handling. Affect of ISR: 0x010C V
(a) The segment and offset of interrupt service routine of int 77h are placed at: offset:	Q1: $[\text{marks } 4x2 + 2x2 + 2 + 2 + 2 + 2 + 2 = 20]_{7} \rightarrow$	-> -> -> -> > segment of Isa: 0x010E
(b) what is the total size of the IVT table? IOGL (1KB) mov es, ax mov [es:400], mySub (d) Which registers are changed by the iret instruction? ID CS SD MOV [es:400], mySub mov [es:402], cs (e) Replace the following independent invalid instructions with a single instruction that has the same effect. (e) Replace the following independent invalid instructions with a single instruction that has the same effect. Solution: POD OC (f) Write a code to swap two registers ax and bx without using temporary space (local variable) on stack or memory. You are only allowed to use stack operations. Solution: POSA OC POSA OC (g) What would be the value of SP after Jmp start	(a) The segment and offset of interrupt service re	outine of int 77h are Code for part (d)
(b) what is the total size of the IVT table? IOGL (1KB) mov es, ax mov [es:400], mySub (d) Which registers are changed by the iret instruction? ID CS SD MOV [es:400], mySub mov [es:402], cs (e) Replace the following independent invalid instructions with a single instruction that has the same effect. (e) Replace the following independent invalid instructions with a single instruction that has the same effect. Solution: POD OC (f) Write a code to swap two registers ax and bx without using temporary space (local variable) on stack or memory. You are only allowed to use stack operations. Solution: POSA OC POSA OC (g) What would be the value of SP after Jmp start	placed at: offset: @wow >> segment:	mov ax, 0
(d) Which interrupt is hooked by the instructions given on right side? mov [es:402], cs (e)Replace the following independent invalid instructions with a single instruction that has the same effect. i) mov ax, [ss:sp] Solution: pop ax ii) sub sp,2 Solution: posh ax (f)Write a code to swap two registers ax and bx without using temporary space (local variable) on stack or memory. You are only allowed to use stack operations. Solution: posh ax pop bx (g)What would be the value of SP after Jmp start	(b) what is the total size of the IVT table? 1024	13 (IKB) moves, ax
(e)Replace the following independent invalid instructions with a single instruction that has the same effect. i) mov ax, [ss:sp]		
(e)Replace the following independent invalid instructions with a single instruction that has the same effect. i) mov ax, [ss:sp] Solution: pop ax ii) sub sp,2 Solution: posh ax f)Write a code to swap two registers ax and bx without using temporary space (local variable) on stack or memory. You are only allowed to use stack operations. Solution: posh ax pop bx (g)What would be the value of SP after Jmp start		s given on right side? mov [es:402], cs
Solution: Pop αα Solution: Pop αα Solution: Pop αα	(_64_)16	taccols !
Solution: Pop αα Solution: Pop αα Solution: Pop αα	le Replace the following independent invalid inst	ructions with a single instruction that has the same
i) mov ax, [ss:sp] add sp, 2 ii) sub sp,2 mov [ss:sp], ax (f)Write a code to swap two registers ax and bx without using temporary space (local variable) on stack or memory. You are only allowed to use stack operations. Solution: \[\text{District Ax} \\ \text{Posh Bx} \\ \text{Pop bx} \] (g)What would be the value of SP after Jmp start	effect.	ructions with a single instruction that has the same
add sp, 2 ii) sub sp,2 mov [ss:sp], ax (f)Write a code to swap two registers ax and bx without using temporary space (local variable) on stack or memory. You are only allowed to use stack operations. Solution: push are poor br poor br (g)What would be the value of SP after Jmp start	The state of the s	Solution:
mov [ss:sp], ax (f)Write a code to swap two registers ax and bx without using temporary space (local variable) on stack or memory. You are only allowed to use stack operations. Solution: \[\text{Dish ax} \\ \text{Posh ax} \\ \text{Posh bx} \\ \text{Qop bx} \] (g)What would be the value of SP after \[\text{Jmp start} \]		bob ag
(f)Write a code to swap two registers ax and bx without using temporary space (local variable) on stack or memory. You are only allowed to use stack operations. Solution: \[\text{Distribution} Dis	ii) sub sp,2	Solution: Out to accompany
Solution: Dish as poop bx (g) What would be the value of SP after Stack or memory. You are only allowed to use stack operations. Solution: Dish as poop bx Dish bs poop by poop bx Dish bs poop by poop bx Dish	mov [ss:sp], ax	posit ad
Solution: push are push branch pop or pop branch pop br		
pop or pop bx (g)What would be the value of SP after Jmp start	stack or memory. You are only allowed to use sta	ick operations.
ρορ σα ρορ bx (g)What would be the value of SP after Jmp start		
(g)What would be the value of SP after	DOD 02	
	pop bx	
	(-NA/hat would be the value of CD after	I lean stort
LOVOCUTION OF THE TOUGHUNG CODE ! DAUTING!	execution of the following code?	Imp start Routine:
Initial value of SP = 0xFFFE Ret		
Solution:		
$SP = O \times FFF \otimes Call routine$		
Push ax OxFFFC		
Sub sp,4 OxFFF8		

(h) Complete the following code to place asterisk '*' character on the left diagonal of the screen.

- 1. Mov ax,0xb800
- 2. Mov es, ax
- 3. Mov ax,0x0742



National University of Computer and Emerging Sciences Lahore Campus



4. Mov cx, 25
5. L1: Mov word [es: si], ax 1 assuming that si = 0 mitially
6. add so 15162
7. Loop 11

Suppose the following declarations have been Solution: made mou si, str I str1: db 'FGHIJ' push ds str2: db 'ABCDE00000 ' pop es Write instructions to move str1 to the end of mov dis sts 2 odd dy 5 str2, producing the string 'ABCDEFGHIJ' using string instructions mov cx, 5 CIG asb worsp

CLO #:2 Describe the working of important x86 assembly primitives, including arithmetic, branching, bit manipulation, addressing modes and interrupt handling.

Q2: Write a subroutine to swap the odd rows with the even ones in the video memory i.e. swap row 0 with row 1. Row 2 with row 3 and so on using string instructions. [20 marks]

Solution:

Swap Odd Even: puch bp mov bp, sp push ar - Can be replaced with push a posh bx push es push di push ds Push si Push cx push dx ; FUNC START mou ex, 1000: *IOOO iterations required since each iterations will deal with two certs (2000/2 = 1000) MM mov ox, oxb800 mov de, ax MOY M S1, 51.

