① Students have either already taken or started taking this quiz, so take care when editing it. If you change any quiz questions in a significant way, you might want to consider re-grading students' quizzes who took the old version of the quiz.

Points 8 Published

:

**Details** 

Questions

✓ Show question details

```
Executing this code gives a segmentation fault.

| Char *src = "hello";
| char *dst;
| strcpy(dst, src);
| Which option can make this code work.

| Any of the given options can work
| Changing the second line to: char dst[6];
| Changing the second line to: char *dst = (char *) malloc(8 * sizeof(char));
| Changing the third line to: strdup(dst, src);
```

```
## Question

Sure, here is a difficult multiple-choice question on malloc() and free():

Which of the following C programs is safe from memory leaks and dangling pointers?

(A)

int main() {
    int *p = malloc(sizeof(int));
    *p = 10;
    return 0;
}

(B)

(b)

(c)

Int main() {
    int *p = malloc(sizeof(int));
    *p = 10;
    p = NULL;
    // ... don't use p ...
    free(p);
    return 0;
}

(c)

Int main() {
    int *p = malloc(sizeof(int));
    *p = 10;
    int *p = malloc(sizeof(int));
    return 0;
}

(d)

Int main() {
    int *p = malloc(sizeof(int));
    int *p = malloc(sizeof(int));
    return 0;
}

(e)

Int main() {
    int *p = malloc(sizeof(int));
    int *p = m
```

```
int main() {
    int *p = malloc(sizeof(int));
    *p = 10;
    // ... use p ...
    free(p);
    // ... use p again ...
    return 0;
}

ISWET

A

B

C

D
```

	Ques	stion					1 pts
	A 16KB virtual memory is mapped to the physical memory as follows:  0KB						
	UNB	Operating System					
	16KB	Operating System					
	TONE	(not in use)					
	32KB						
		Code Heap   (allocated but not in use)					
	48KB	† Stack					
		(not in use)					
	64KB						
The virtual address 15 KB maps to which physical address?							
wer							
vei	0_4	7KB					
	O 3	3KB					
	O 3	1KB					
	O 6	3KB					

∄ Qu	uestion	1 pts
	ranslating virtual addresses to physical addresses, we need:	
` '	ase/bound registers pair	
(B) ab	bility to translate virtual addresses and check if within bounds	
, ,	bility to translate virtual addresses and check if within bounds h of the following is true?	
Which		
Which	h of the following is true?	

(A) is provided by the OS, (B) is provided by the hardware

	ii Question			
	Suppose a 32 KB virtual address space is divided into 4 equal segments. Here is the logic to translate it to the physical address:  Segment = (VirtualAddress & SEG_MASK) >> SEG_SHIFT  // now get offset Offset = VirtualAddress & OFFSET_MASK if (Offset >= Bounds[Segment])     RaiseException(PROTECTION_FAULT) else     PhysAddr = Base[Segment] + Offset     Register = AccessMemory(PhysAddr)  Then which of the following is true?			
ıswer	○ 3EG_NIASK - 0x0000, 3EG_SHIFT - 13, OFFSET_NIASK - 0x1FFF			
	○ SEG_MASK = 0x3000, SEG_SHIFT = 12, OFFSET_MASK = 0xFFF			
	<ul><li>○ SEG_MASK = 0xC000, SEG_SHIFT = 14, OFFSET_MASK = 0x3FFF</li><li>○ SEG_MASK = 0x1800, SEG_SHIFT = 11, OFFSET_MASK = 0x7FF</li></ul>			
	Which of the following is true:	1 pts		
ıswer	Segmentation helps avoid internal fragmentation, but not external fragmentation			
	Segmentation helps avoid external fragmentation, but not internal fragmentation			
	Segmentation helps avoid both internal and external fragmentation			
	Segmentation is neither able to deal with internal nor external fragmentation			
	ii Question	1 pts		
	One message queue allows communication between:	<b>⊗</b> ×		
ıswer	multiple senders and multiple receivers			
	O one sender and one receiver			

iii Question 1 pts

A shared memory between two processes is at:

one sender and multiple receivers

multiple senders and one receiver

swer	O different virtual addresses but the same physical address		
	same virtual and physical addresses		
	<ul> <li>same virtual address but different physical addresses</li> </ul>		
	O different virtual addresses and different physical addresses		
	+ New question + New question group Q Find questions		
	Notify users this quiz has changed		
		Cancel	Save