

四川大学期末考试试题（闭卷）

（2018--2019 学年第 1 学期）

A 卷

课程号： 311046040 课程名称： 系统级编程 任课教师： _____

适用专业年级： 软件工程 2016 级 学号： _____ 姓名： _____

考生承诺							
我已认真阅读并知晓《四川大学考场规则》和《四川大学本科学考试违纪作弊处分规定（修订）》，郑重承诺：							
1、已按要求将考试禁止携带的文具用品或与考试有关的物品放置在指定地点；							
2、不带手机进入考场；							
3、考试期间遵守以上两项规定，若有违规行为，同意按照有关条款接受处理。							
考生签名： _____							
题 号	一(40%)	二(10%)	三(5%)	四(15%)	五(12%)	六(8%)	七(10%)
得 分							
卷面总分			阅卷时间				

- 注意事项：**
1. 请务必将本人所在学院、姓名、学号、任课教师姓名等信息准确填写在试题纸和添卷纸上；
 2. 请将答案全部填写在本试题纸上；
 3. 考试结束，请将试题纸、添卷纸和草稿纸一并交给监考老师。
-

评阅教师	得分

一、单项选择题（本大题共 20 小题，每小题 2 分，共 40 分）

提示：在每小题列出的四个备选项中只有一个是符合题目要求的，请将其代码填写在下表中。错选、多选或未选均无分。

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20

1. Which one can translate assembly into machine code?
 - a. Debugger
 - b. Assembler
 - c. Loader
 - d. Linker
2. Which expression is false?
 - a. $!(0)$
 - b. $\sim 0xFF$
 - c. $\sim 0x00$
 - d. $0x00 \wedge 0x00$

3. Given the following code segments, which statement can NOT alter 'b' in str1 to 'd'?
- ```
char str1[]="abc"; char *p;
```
- str[1] = 'd';
  - \*(p++) = 'd';
  - \*(str1++) = 'd';
  - p=str1; p[1]='d';
4. Which statement has NO effect on stack?
- MOV
  - POP
  - PUSH
  - CALL
5. The Visual C++ Memory window displays
- only local variables allocated in stack, without the associated variable names
  - both local and global variables allocated in stack, without the associated variable names
  - the contents of memory, interpreted in one of several ways, without the associated variable names
  - the contents of memory, interpreted as 32-bit integers, without the associated variable names
6. What happens in a C program when an addition would cause integer overflow?
- The correct value is coerced to a floating point number.
  - Execution is terminated.
  - An incorrect result is produced and execution continues.
  - An exception-handler is called with the two operands as parameters.
7. The C expression a->b is equivalent to
- (&a) + b
  - (\*a).b
  - (&a).b
  - \*(a + b)
8. To quickly allocate and free many variables of a commonly used data type, we could
- use sizes which are powers of two.
  - keep a linked list of free objects of that type's size.
  - minimize the size of the data type.
  - coalesce blocks when they are freed.

9. The program counter contains
- the number of times a program has been executed
  - the address of the CPU instruction that is about to be executed
  - the amount of memory a program is currently using
  - the number of CPU instructions a program has executed so far
10. Assume a function foo takes two arguments. When calling foo(arg1, arg2), which is the correct order of operations assuming x86 calling conventions and that foo must allocate stack space(implies that we must save the ebp)?
- push arg1, push arg2, call foo, push ebp
  - push arg1, push arg2, push ebp, call foo
  - push arg2, push arg1, call foo, push ebp
  - push arg2, push arg1, push ebp, call foo
11. Which of the following statements about alignment within C struct's is true?
- Alignment may cause the allocation of unused space.
  - Alignment is required by all modern processors.
  - Alignment can help processors access data more efficiently.
- i and iii only
  - i only
  - ii and iii only
  - i, ii, and iii
12. Consider the following code.
- ```
char a[100];  
a[99] = *((char *) (((int) &a[0]) + 4))
```
- If integers are 32 bits wide, which of the following values is equal to a[99]?
- a[0] + 4
 - a[3]
 - a[4]
 - the integer stored in the bytes a[4], a[5], a[6] and a[7]
13. Suppose that, using a tool such as the memory window of Visual C++, we found that a certain set of contiguous memory locations contained the integer 0xC605CD623A8365000000. What could these memory locations hold?
- the integer 0xC605CD623A8365000000
 - a string
 - a CPU instruction

- a. i only
- b. iii only
- c. i, ii, and iii
- d. i and ii only

14. Two code fragments are i) and ii) as following:

```
i) M= 100000; N=100000;
int data[M][N];
for (i = 0; i < M; i++) {
    for (j = 0 ; j < N; j++) {
        sum += data[ i ] [ j ];
    }
}
```

```
ii) M= 100000; N=100000;
int data[M][N];
for (j = 0 ; j < N; j++) {
    for (i = 0; i < M; i++) {
        sum += data[ i ] [ j ];
    }
}
```

which one is more efficient (better performance) if you program with Matlab which is column-major order for matrix?

- a. Both are very efficient.
- b. i)
- c. ii)
- d. Both are not efficient.

15. the advantage of using copying gc including

- i) A copying collector is generally more efficient than a non-copying collector
 - ii) The copying gc can make use of heap memory effectively.
- a. i)
 - b. ii)
 - c. both of them
 - d. none of them

16. which of the following are advantages of using statistical sampling to profile programs?

- i) exact run times of all functions can be determined.
 - ii) code can be instrumented automatically.
 - iii) the performance impact due to measurement can be minimal
- a. i) and iii) only

- b. ii) and iii) only
- c. i) and ii) only
- d. i), ii), and iii)

17. read the following code, and how can we optimize it?

```
void lower1(char *s) {  
    int i;  
    for (i = 0; i < strlen(s); i++)  
        if (s[i] >= 'a' && s[i] <= 'z')  
            s[i] -= ('a' - 'a');  
}
```

- a. reducing procedure calls
- b. enhancing parallelism
- c. loop splitting
- d. converting to pointer code

18. Which of the following manages the transfer of data between the CPU registers and the cache?

- a. Registry.
- b. Compiler.
- c. Hardware.
- d. Operating System.

19. What can Loader do?

- i) translate the C code into machine code
 - ii) Resolution
 - iii) load or map the Executable object file from the disk to memory
- a. i) and ii) only.
 - b. i) and iii) only.
 - c. i), ii) and iii).
 - d. iii) only.

20. In IA32 or X86, which exception never returns back to the point where exception happens?

- a. interrupt
- b. trap
- c. fault
- d. Abort

评阅教师	得分

二、bit operation (本大题共 2 小题, 各 5 分, 共 10 分)

Now complete the following functions according to the following rules.

Each "Expr" is an expression using ONLY the following:

1. Integer constants 0 through 0xFFFFFFFF inclusive.
2. Function arguments and local variables (no global variables).
3. Some of the problems restrict the set of allowed operators.

You are expressly forbidden to:

1. Use any control constructs such as if, do, while, for, switch, etc.
2. Define or use any macros.
3. Define any additional functions in this file.
4. Call any functions.
5. Use any other operations, such as &&, ||, -, ?, or [] :
6. Use any form of casting.

You may assume that your machine:

1. Uses 2s complement, 32-bit representations of integers.
2. Performs right shifts arithmetically.
3. Has unpredictable behavior when shifting an integer by more than the word size.

/*

* isNonNegative - return 1 if x >= 0, return 0 otherwise

* Example: isNonNegative(-1) = 0. isNonNegative(0) = 1.

* Legal ops: ! ~ & ^ | + << >>

* Max ops: 6

* Rating: 3

*/

int isNonNegative(int x) {

}

/*

* isEqual - return 1 if x == y, and 0 otherwise

* Examples: isEqual(5,5) = 1, isEqual(4,5) = 0

* Legal ops: ! ~ & ^ | + << >>

* Max ops: 5

* Rating: 2

*/

```
int isEqual(int x, int y) {
```

```
}
```

评阅教师	得分

三、memory allocation (本大题共1小题, 共5分)

Are there any errors in the following program? If there are, please correct them on the right side.

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
char* getmemory()
```

```
{
```

```
    char *p=(char *) malloc(100);
```

```
    strcpy(p,"hello world");
```

```
    return &p;
```

```
}
```

```
int main( )
```

```
{
```

```
    char *str=NULL;
```

```
    str = getmemory( );
```

```
    printf("%s/n",str);
```

```
    free(str);
```

```
    return 0;
```

```
}
```

评阅教师	得分

四、stack discipline (本大题共2小题, 共15分)

Consider the following C code and assembly code for a recursive function:

```
int gcd(int a, int b)
```

```
{
    if(!b) {
        return a;
    }
    return gcd(b, a % b);
}
```

```
00401020    push    ebp
00401021    mov     ebp,esp
00401023    sub     esp,40h
.....
00401038    cmp     dword ptr [ebp+0Ch],0
0040103C    jne     gcd+23h (00401043)
0040103E    mov     eax,dword ptr [ebp+8]
00401041    jmp     gcd+37h (00401057)
00401043    mov     eax,dword ptr [ebp+8]
00401046    cdq
00401047    idiv    eax,dword ptr [ebp+0Ch]
0040104A    push    edx
0040104B    mov     eax,dword ptr [ebp+0Ch]
0040104E    push    eax
0040104F    call    @ILT+0(_gcd) (00401005)
00401054    add     esp,8
00401057    pop     edi
00401058    pop     esi
00401059    pop     ebx
.....
00401064    mov     esp,ebp
00401066    pop     ebp
00401067    ret
```

Imagine that a program makes the procedure call `gcd(213, 18)`. Also imagine that prior to the invocation, the value of `esp` is `0xffff1000`—that is, `0xffff1000` is the value of `esp` immediately before the execution of the call instruction.

1. (12 points) Note that the call `gcd(213, 18)` will result in the following function invocations: `gcd(213,18)`, `gcd(18, 15)`, `gcd(15, 3)`, and `gcd(3, 0)`. Using the provided code and your knowledge of IA32 stack discipline, fill in the stack diagram with the values that would be present immediately before the execution of the leave instruction for `gcd(15, 3)`. Supply numerical values wherever possible, and cross out each blank for which there is insufficient information to complete with a numerical value.

Address no.	Value	Description
0xffff1008		
0xffff1004		
0xffff1000		
0xffff0ffc		
0xffff0ff8		
0xffff0ff4		
0xffff0ff0		

Address no.	Value	Description
0xffff0fec		
0xffff0fe8		
0xffff0fe4		
0xffff0fe0		
0xffff0fdc		
0xffff0fd8		
0xffff0fd4		
0xffff0fd0		
0xffff0fcc		
0xffff0fc8		
0xffff0fc4		
0xffff0fc0		
0xffff0fbc		
0xffff0fb8		
0xffff0fb4		
0xffff0fb0		

2. (3 points) What are the values of esp and ebp immediately before the execution of the ret instruction for gcd(15, 3)?

评阅教师	得分

五、performance optimization (本大题共 2 小题, 共 12 分)

1. (6 points) We are considering an enhancement to the processor of a web server. Then new CPU is 10 times faster on search queries than the old processor. The old processor is busy with search queries 60% of the time, what is the speedup gained by integrating the enhanced CPU?

2. (6 points) Read the following two code fragments, func1(x) and func2(x) . Do func1(x) and func2(x) have same behavior in any circumstances? Give an example of f(int) to show that func1(x) and func2(x) have the same behavior. If their behaviors are not same in some cases, give another example of f(int).

```
int f(int);
int func1(x) {
    return f(x) + f(x) + f(x) + f(x);
}
int func2(x) {
    return 4*f(x);
}
```

评阅教师	得分

六、cache（本题共8分）

Read the code as following:

```
int i
int sum=0;
int data[100000];
for (i = 0; i < 100000; i += stride)
    sum += data[i];
```

Assume that size of int is 4-byte and the cache line (block) length is 32-byte.

If the stride is 1,2,4,8 , what is their hit ratio and miss ratio? Fill the blanks in the table and give your explain.

Stride	Hit ratio	Miss ratio
1		
2		
4		
8		

评阅教师	得分

七、linker (本题共 10 分)

There is a program built by the following two c files: main.c and swap.c. Reading the source code, and please give a description on when linking, what should linker do in order to merge them. And drawing a figure is suggested.

```
/*main.c*/
extern int shared;
int main()
{
    int a=100;
    swap(&a,&shared);
}
/*swap.c*/
int shared=1;
void swap(int* a, int* b)
{
    *a^=*b^=*a^=*b;
}
```