四川大学期末考试试题（闭卷）

# （2017~2018 学年第 1 学期） A卷

课程号： **311143040** 课程名称：系统级编程任课教师：

适用专业年级：软件工程 **2015** 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《四川大学本科学生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**(13%)** | 四**(12%)** | 五**(15%)** | | 六**(10%)** |
| 得分 |  |  | |  |  |  | |  |
| 卷面总分 |  | | 教师签名 |  | | 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；

1. 请将答案全部填写在本试题纸上；
2. 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 of the following languages is hardware dependent?
   1. C
   2. Assembly
   3. C++
   4. JAVA

答案：B，汇编语言依赖于硬件，与硬件有很大的关系。

1. Which expression is false?
   1. !(-5)
   2. ~0x01
   3. ~0x00
   4. !0

答案：A，除了0之外的任意数字，逻辑上的非都是0，也就是false。

1. Given the following code segments, which statement is NOT true?

Statement 1: char str1[]=“abc”;

Statement 2: char \*str2 =“abc”;

Statement 3: char \*str3 = “abc”

* + 1. statement 1 requires 3 bytes from stack
    2. statement 2 requires 4 bytes from stack
    3. expression (str2 = =str3) is true
    4. expression (str1 = =str2) is true
  1. II III and IV B. II and III

C. I and IV D. II and IV

答案：C，三个字符需要四个字节的空间，因为末尾有个’\0’,字符串”abc”是常量，在静态区域，所以str2和str3指向的是同一个位置，而str1不是，所以IV不正确，故结果选C。

1. The machine code generated from source code by a compiler
   1. executes more quickly than the source code
   2. does not preserve all the information given in the source code
   3. can be easily inspected to check the correctness of the compiler
   4. associates variable values with their names

答案：B，源代码中的所有变量名，函数名还有注释这些信息编译器产生的机器代码都是不保留的，所以B是对的。A选项比源代码更快，无优化的情况应该一样。编译器的正确性与从源代码产生机器代码无关的吧。

1. After execution of the following code in a 32 bits system, which statement is

TRUE?

char a[8]={‘1’,’2’,’3’,’4’,’5’,’6’,’7’,’8’}; double \*p = (double \*)&a;

\*p= -1.5f;

A. a[0] is ‘1’ B. a[7] is 0

C. a[8] is 1 D. None of the above

答案：B，-1.5的二进制形式是11.1最左边是符号位，那么其尾数只有第一位是0，则说明后面几个字节全是0，故a[7]为0，答案选B，第一个字节肯定不是1，符号位是1，起码大于128，具体多少需要看如何表示双精度，但肯定的不是1.

1. In C, local variables allocated inside functions are allocated
   1. on the stack B. in static storage

C. in the heap D. in a fifo

答案：A，局部变量分配在栈上

1. Which statement will copy the value in register BX to AX?
   1. LDS AX,BX B. LEA AX,BX

C. MOV AX,BX D. STOS AX,BX

答案：C，mov指令是用来复制值的，但是好像复制的情况恰好与题上相反，因为此处是将AX的值赋给了BX，恰恰与题上所说相反。

1. Consider the following segment of C source code.

int a = 8; int b = \*&a;

What is the value of variable b at the end of execution of the segment?

* 1. a B. (int) &a

C. (int) &b D. &a

答案：A，&a取的a的地址，\*(&a)指的就是该地址上的内容，即为a的值：8.

1. A memory leak is caused by a

A.failure to free allocated memory

B.bug in which too much memory is allocated, causing internal fragmentation

C.bug in the memory allocator that fails to free memory

D.function that allocates a large amount of memory from the heap

答案：C，释放内存失败。

1. What properties of a variable are specified by the static keyword in c?

i.the variable will be statically allocated.

ii.the variable name will be visible only to functions defined within the same file.

iii.the variable's value does not change very often. the compiler uses this fact to focus optimizations on other variables.

A. i only B. i and iii only.

C. i and ii only. D. iii only

答案：A，变量加了static修饰，全局变量改变作用域变为只有该文件能访问，局部变量改变生存周期，不会调用完，就无法使用。

1. We use dynamic memory because:
   * 1. The heap is significantly faster than the stack.
     2. The stack is prone to corruption from buffer overflows.
     3. Storing data on the stack requires knowing the size of that data at compile time.
     4. None of the above.

答案：C，动态分配就是说目前大小是不确定的，而栈分配必须确定大小，所以选C

1. Which one of the following statements is NOT true of storage allocators?
   * 1. In the best case, coalescing with boundary tags is linear in the number of free blocks.
     2. Segregated free list typically approximates best fit search.
     3. Payloads must be aligned to some boundary.
     4. Explicit lists are typically faster than implicit list.

答案：A，询问的表哥，网上我没有查到结果。

1. A garbage collector starts from some "root" set of places that are always considered "reachable", such as

i. CPU registers ii. stack

iii. global variables

A. i & ii B. i & iii

C. iii D. all of them

答案：D，上述全部都是，记住就行了。

1. Which on the following is a reason next-fit might perform better than first-fit?
   * 1. If a large number of small blocks are stored at the beginning of the free list, next-fit avoids walking through those small blocks upon every allocation.
     2. First fit requires a traversal of the entire free list, but next-fit does not.
     3. First-fit requires that both allocated and unallocated blocks be examined, and next-fit examines only free blocks.
     4. Next-fit is an approximation of best-fit, so it reduces internal fragmentation compared to first-fit.

答案：A，next-fit是接着上次分配的开始找第一个符合大小的块，first-fit是每次都从第一块开始找合适的那一块。当开头有很多小块时，next-fit就能跳过那些，而first-fit还是要比对那些小块，此时next-fit更有效。

1. Which of the following cannot be used to estimate time consumption of a program?
   * 1. RDTSC
     2. QueryPerformanceFrequency
     3. GetTickCount
     4. clock() / CLOCKS\_PER\_SEC

答案：B，返回的是一个频率。

1. from the time when a c program is written, to the time when it is running as a process on windows, what should be done?

i.compile ii.link iii.load.

A.i and ii only.

B.i and iii only.

C.i, ii and iii.

D.ii only.

答案：C，从写c代码到运行，涉及到编译、链接、加载，故全部都有。

1. a lock is a software mechanism that

A.temporarily makes memory read-only.

B.limits access to a critical section.

C.implements password protection to data.

D.prevents execution except in debug mode.

答案：B，锁是为了限制访问共同使用的资源

1. which of the following is not optimization technique?

A.code motion

B.loop unrolling

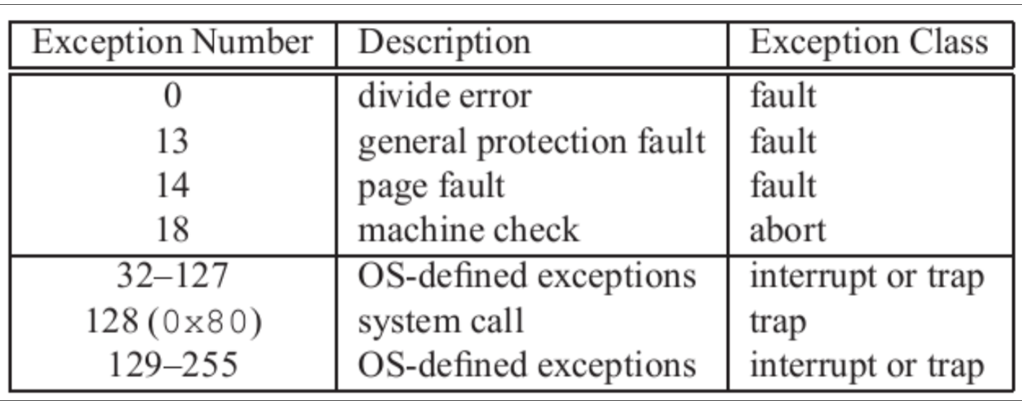
C.constant folding

D.memory aliasing

答案：D，内存别名是阻止的一个因素。

1. in ia32 or x86, which exception is used to implement system call
   * 1. interrupt
     2. trap
     3. fault
     4. Abort

答案：B



1. which facts about the cache can be determined by calling the following function?

int data[1 << 20]; void callee(int x) { int i, result;

for (i = 0; i < (1 << 20); i += x) {

result += data[i];

} }

i cache line size

ii cache size

iii cache speed

* + 1. i and ii only
    2. i only
    3. i and iii only
    4. i, ii, and iii

答案：B，cache里放的数据多命中率就高，效率就高。

|  |  |
| --- | --- |
| 评阅教师 | 得分 |
|  |  |

二、bit operation（本大题共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:
4. Use any control constructs such as if, do, while, for, switch, etc.
5. Define or use any macros.
6. Define any additional functions in this file.
7. Call any functions.
8. Use any other operations, such as &&, ||, -, ?, or [] :
9. 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.

/\* NegativeNum using only ~ and & , ignore 0

* Example: NegativeNum(-5) retrun -5 , NegativeNum(5) retrun -5, Negative(0) can return any value \* Legal ops: ~ & \* Max ops: 8 \*/

int NegativeNum (int x)

{

return ~(~(~x+1)&~x);

}

/\*

* SetByte - Set byte n from word x to 0xFF
* Bytes numbered from 0 (LSB) to 3 (MSB)
* Examples: SetByte(0x12345678,1) = 0x1234FF78
* Legal ops: ! ~ & | << >> \* Max ops: 6\*/

int SetByte (int x,int n)

{

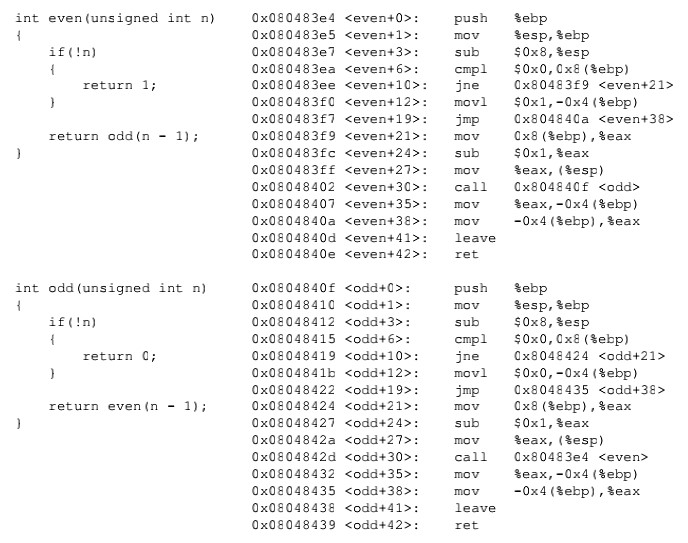
return x | 0x000000FF << n \* 8;

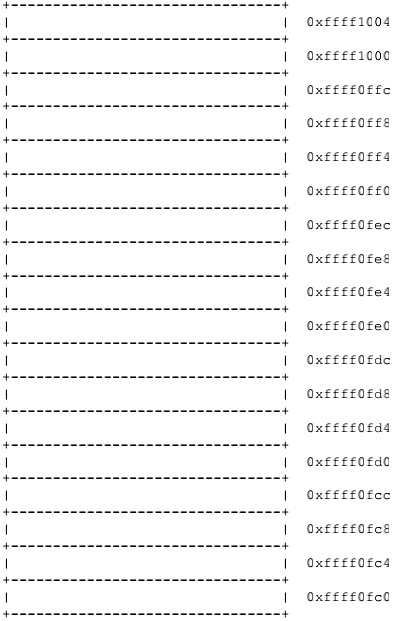
}

|  |  |
| --- | --- |
| 评阅教师 | 得分 |
|  |  |

三、stack discipline（本大题共13 分）。

Consider the following C code and assembly code for two mutually recursive functions:



Imagine that a program makes the procedure call **even(3)**. 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. Note the the call even(3) will result in the following function invocations: even(3), odd(2), even(1), and odd(0). Full in the stack diagram with the values that would be present immediately before the execution of the ret instruction for odd(0). Cross out each blank for which there is insufficient information to complete.

04 unknown

00 3

fc 返回地址

f8 old ebp

f4

f0

ec 2

e8 返回地址

e4 f8

e0

dc

d8 1

d4 返回地址

d0 e4

cc

c8

c4 0

c0 返回地址

bc d0

b8

b4

2. What are the values of ESP and

EBP immediately before the execution of the ret instruction for odd(0)?

ESP= \_\_\_\_\_\_\_\_\_\_\_ffff0fc0\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

EBP=\_\_\_\_\_\_\_\_\_\_\_ ffff0fd0\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| 评阅教师 | 得分 |
|  |  |

四、Performance Optimization（本大题共12 分）。

1. Consider the following functions:

int min(int x, int y) {

return x < y ? x : y;

}

int max(int x, int y) {

return x < y ? y : x;

}

void incr(int \*xp, int v) {

\*xp += v;

}

int square(int x) {

return x \* x;

}

Here are three code fragments that call these functions

1. for (i = min(x, y); i < max(x,y); incr(&i, 1)) { t += square(i); }
2. for (i = max(x, y) - 1; i >= min(x,y); incr(&i, -1)) { t += square(i); }
3. int low = min(x, y); int high = max(x, y);

for (i = low; i < high; incr(&i, 1)) { t += square(i); }

assume x = 10 and y = 100. Fill in the table below indicating the number of times each of the four functions is called for each of these code fragments.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Code | min | max | incr | square |
| A | 1 | 91 | 90 | 90 |
| B | 91 | 1 | 90 | 90 |
| C | 1 | 1 | 90 | 90 |

2. The following problem illustrates the way memory aliasing can cause unexpected program behavior. Consider the following procedure:

// Swap value x at xp with value y at yp void Swap(int \*xp, int \*yp)

{

\*xp = \*xp + \*yp; // x + y

\*yp = \*xp - \*yp; // x+y-y=x

\*xp = \*xp - \*yp; // x+y-x=y

}

Can this procedure be used to swap two values? And why?

解析：不能，当都指向同一个位置时，就不行。

|  |  |
| --- | --- |
| 评阅教师 | 得分 |
|  |  |

五、Cache（本大题共15 分）。

1. Analyse the locality the following code with nested loop. Note：This is the [pseudo](https://cn.bing.com/dict/search?q=pseudo&FORM=BDVSP6&mkt=zh-cn)[-code.](https://cn.bing.com/dict/search?q=code&FORM=BDVSP6&mkt=zh-cn)

integer data(M, N); （5分 ） for (i = 0; i < M; i=i+1) {

for (j = 0 ; j < N; j=j+1) { sum += data( i , j );

}

}

数组一行一行顺着排下去的，访问一个时，加入附近的几个，下一次就会访问

2.You are evaluating a system’s cache performance on a machine with a 1024-byte direct-mapped data cache with 16-byte blocks (B =16). You are given the following definitions: （10分 ）

struct position {

int x;

int y;

int z1;

int z2;

};

struct position grid[16][16]; int total\_x = 0, total\_y = 0, total\_z = 0; int i, j, k;

You should also assume:

* sizeof(int) == 4.
* grid begins at memory address 0.
* The cache is initially empty.
* The only memory accesses are to the entries of the array grid. Variables i, j , total\_x, total\_y and total\_z are stored in registers.

Determine the cache performance of the following code: for (i = 0; i < 16; i++)

{

for (j = 0; j < 16; j++)

{

total\_x += grid[i][j].x; total\_y += grid[i][j].y;

total\_z += grid[i][j].z1+ grid[i][j].z2;

}

}

1. What is the total number of reads? \_\_\_16\*16\*4=1024\_\_\_\_.
2. What is the total number of reads that miss in the cache? \_\_\_256\_\_\_\_ .
3. What is the miss rate? \_\_\_0.25\_\_\_\_.
4. What would the miss rate be if the cache were twice as big? And explain every answer briefly.

答案：0.125

|  |  |
| --- | --- |
| 评阅教师 | 得分 |
|  |  |

六、Answering Question（本大题共10 分）。

1. What is the mean of our textbook of CSAPP？Please give your understanding about it, such as what you have learned from this course or this textbook.（5分）

开放性，不作答。

1. What is exception? Please list the four types of exception in IA32, and draw a figure to show the different.（5分）

interrupt：异步 总是返回下一条指令

trap：同步 总是返回下一条指令

fault：同步 返回当前指令

abort：同步 不返回任何指令