

## 2022 年春季学期计算机系统试题（A）参考答案

### 一、单项选择题（每题 2 分，共 20 分）

题号	1	2	3	4	5
答案	D	B	C	B	B

题号	6	7	8	9	10
答案	C	D	D	D	C

### 二、填空题（每空 2 分，共 20 分）

题号	答案	分值
1	0000FFFAH	2 分
2	16	2 分
3	C480 0000	2 分
4	4/B	2 分
5	0x00b	2 分
6	0	2 分
7	符号解析	2 分
8	页表	2 分
9	MMU/内存管理单元	2 分
10	13.35	2 分

### 三、简答题（总分 15 分，意思对即可酌情给分）

1. 简述 Y86-64 流水线 CPU 中的冒险的种类与处理方法。

答：（3 分）数据冒险：指令使用寄存器 R 为目的，瞬时之后使用 R 寄存器为源。处理方法有暂停： 通过在执行阶段插入气泡（bubble/nop），使得当前指令执行暂停在译码阶段；数据转发：增加 valM/valE 的旁路路径，直接送到译

码阶段；加载使用冒险：指令暂停在取指和译码阶段，在执行阶段插入气泡（bubble/nop）

**(2 分)** 控制冒险：分支预测错误：在条件为真的地址 target 处的两条指令分别。插入 1 个 bubble。 ret：在 ret 后插入 3 个 bubble。

2. 下列 C 程序存在安全漏洞，请给出攻击方法。如何修复或防范？

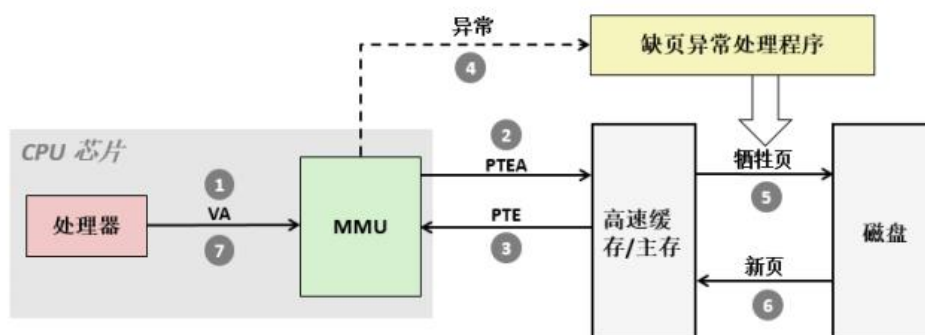
```
int getbuf(char *s) {  
    char buf[32];  
    strcpy(buf, s);  
}
```

**答：(3 分)** 攻击方法：采用基于缓冲区溢出攻击，让输入字符串 s 的字符个数大于 32，可导致 getbuf 函数返回到无关的代码处，或返回到指定的攻击代码处。

**(2 分)** 修复：限制字符串操作的长度可编码缓冲区溢出的攻击，如：

```
int getbuf_s(char *s) {  
    char buf[32];  
    if(strlen(s)<sizeof(buf))  
        strcpy( buf, s );  
}
```

3. 结合下图，简述虚拟内存地址翻译的过程。



**答：**(1) 处理器将虚拟地址发送给 MMU----- (0.5 分)

(2)、(3) MMU 利用虚拟地址对应的虚拟页号生成页表项（PTE）地址，并从页表中找到对应的 PTE----- (0.5 分)

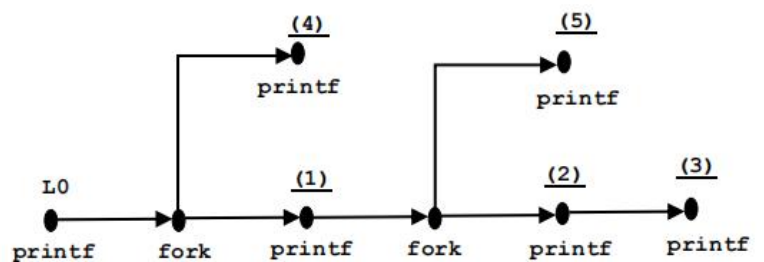
(4) PTE 中的有效位为 0，MMU 触发缺页异常----- (1 分)

- (5) 缺页处理程序选择物理内存中的牺牲页（若页面被修改，则换出到磁盘）  
-----（1分）
- (6) 缺页处理程序调入新的页面到内存，并更新 PTE-----（1分）
- (7) 缺页处理程序返回到原来进程，再次执行导致缺页的指令-----（1分）

#### 四、系统分析题（共 10 分）

1. C 程序 forkB 的源程序与进程图如下：（5 分）

```
void forkB()
{
    printf("L0\n");
    if(fork() !=0) {
        printf("L1\n");
        if(fork() !=0) {
            printf("L2\n");
        }
    }
    printf("Bye\n");
}
```



请写出上述进程图中空白处的内容

(1)   L1   (2)   L2   (3)   Bye    
 (4)   Bye   (5)   Bye  

2. C 程序(64 位模式)的 main 函数参数 argv 地址为 0x0000413433323110,其内容如下：（5 分,各个位置分别是 1 分、2 分、2 分）

0x0000413433323110: 30 31 32 33 34 41 00 00 33 31 32 33 34 41 00 00  
 0x0000413433323120: 35 31 32 33 34 41 00 00 00 00 00 00 00 00 00 00  
 0x0000413433323130: 30 43 00 30 00 32 42 00 38 00 31 31 32 32 00 30  
 0x0000413433323140: 32 33 00 61 41 00 31 00 32 00 33 00 31 00 00 31

请写出 程序名:       0C      , 本程序的参数个数   2   按顺序  
 写出各个参数为       0         2B  

提示: int main(int argc, char \*argv[]); 字符 0、A、a 的 ASCII 为 0x30、  
 0x41、0x61

## 五、综合分析题（共 35 分）

1. 请首先在下表第一列按顺序写出 Y86 指令的各个阶段，然后表中写出 Y86-64CPU 顺序结构设计中 `mrmovq` 和 `ret` 指令各阶段的微操作。

并为 Y86-64 CPU 增加一条指令“`mraddq D(rB), rA`”，能够将内存数据加到寄存器 `rA`。请参考 `mrmovq`、`addq` 指令，合理设计 `mraddq D(rB), rA` 指令在各阶段的微操作，

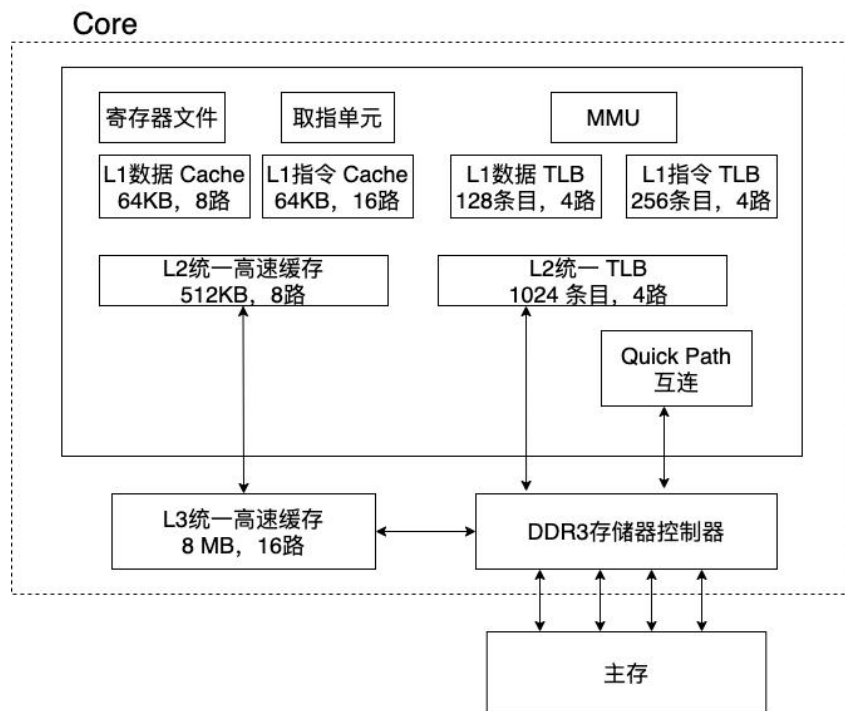
(每列 4 分, 总计 16 分)

指令 `mraddq D(rB), rA` 的编码规则如下

字节 0		字节 1		字节 2...9
C	0	rA	rB	D

	<code>mrmovq D(rB), rA</code>	<code>ret</code>	<code>mraddq D(rB), rA</code> (答案 1)	<b><code>mraddq D(rB), rA</code></b> (答案 2)
取指	$\text{icode:ifun} \leftarrow \text{M1}[\text{PC}]$ $\text{rA:rB} \leftarrow \text{M1}[\text{PC}+1]$ $\text{valC} \leftarrow \text{M8}[\text{PC}+2]$ $\text{valP} \leftarrow \text{PC}+10$	$\text{icode:ifun} \leftarrow \text{M1}[\text{PC}]$  $\text{valP} \leftarrow \text{PC}+1$	$\text{icode:ifun} \leftarrow \text{M1}[\text{PC}]$ $\text{rA:rB} \leftarrow \text{M1}[\text{PC}+1]$ $\text{valC} \leftarrow \text{M8}[\text{PC}+2]$ $\text{valP} \leftarrow \text{PC}+10$	$\text{icode:ifun} \leftarrow \text{M1}[\text{PC}]$ $\text{rA:rB} \leftarrow \text{M1}[\text{PC}+1]$ $\text{valC} \leftarrow \text{M8}[\text{PC}+2]$ $\text{valP} \leftarrow \text{PC}+10$
译码	$\text{valB} \leftarrow \text{R}[\text{rB}]$	$\text{valA} \leftarrow \text{R}[\%rsp]$ $\text{valB} \leftarrow \text{R}[\%rsp]$	$\text{valA} \leftarrow \text{R}[\text{rA}]$ $\text{valB} \leftarrow \text{R}[\text{rB}]$	$\text{valA} \leftarrow \text{R}[\text{rA}]$ $\text{valB} \leftarrow \text{R}[\text{rB}]$
执行	$\text{valE} \leftarrow \text{valB} + \text{valC}$	$\text{valE} \leftarrow \text{valB} + 8$	$\text{valE} \leftarrow \text{valB} + \text{valC}$ $\text{ValM} \leftarrow \text{M8}[\text{ValE}]$ $\text{valE} \leftarrow \text{valM} + \text{valA}$ Set CC	<b><math>\text{valE} \leftarrow \text{valB} + \text{valC}</math></b>
访存	$\text{ValM} \leftarrow \text{M8}[\text{ValE}]$	$\text{ValM} \leftarrow \text{M8}[\text{ValA}]$		<b><math>\text{ValM} \leftarrow \text{M8}[\text{ValE}]</math></b> <b><math>\text{valE} \leftarrow \text{valM} + \text{valA}</math></b> <b>Set CC</b>
写回	$\text{R}[\text{rA}] \leftarrow \text{valM}$	$\text{R}[\%rsp] \leftarrow \text{valE}$	$\text{R}[\text{rA}] \leftarrow \text{valE}$	$\text{R}[\text{rA}] \leftarrow \text{valE}$
更新 PC	$\text{PC} \leftarrow \text{valP}$	$\text{PC} \leftarrow \text{valM}$	$\text{PC} \leftarrow \text{valP}$	$\text{PC} \leftarrow \text{valP}$

2. （8分）某处理器的虚拟地址为 32 位。虚拟内存的页大小是 4KB，物理地址为 48 位，Cache 块大小为 32B。物理内存按照字节寻址。其内部结构如下图所示，依据这个结构，回答下面几个问题：



（1）L1 数据 Cache 有多少组？相应的 Tag 位，组索引位和块内偏移位分别是多少？

答：有256组，Tag有35位，组索引位 8 位，块内偏移 5 位。

（2）对于某数据，其访问的虚拟地址为 0x829358B，则该地址对应的 VPO 为多少？对应的 L1 TLBI 位为多少？（用 16 进制表示）

答：所以组索引 5 位，VP0为 0x58B ，组索引对应的位为 0x13。

（3）对于某指令，其访问的物理地址为 0x829358B，则该地址访问 L1 Cache 时，CT 位为多少？CO 位为多少？（用 16 进制表示）

答：CT 位为 0x8293 CO 位为 0x0B。

3. 列举几种程序优化的方法，并简述其原理。（至少 2 种）

程序优化：矩阵  $c[n][n] = a[n][n] * b[n][n]$ ，请针对该程序进行优化，写出优化后的程序，并说明优化的依据。（11 分）

```
int i, j, k;
for(i = 0; i < n; i++)
{
    for(j = 0; j < n; j++)
    {
        C[i][j] = 0;
        for(k = 0; k < n; k++)
            C[i][j] += A[i][k] * B[k][j];
    }
}
```

答：（至少 2 种，每种 2 分，最多 4 分）

- 1) 消除不必要的内存引用，减少访存次数。
- 2) 用简单操作替代复杂操作，如移位、加替代乘法/除法。
- 3) 共享公共子表达式，重用表达式的一部分。
- 4) 减少过程调用，消除不必要的跳转指令。
- 5) 循环展开，充分利用计算机的功能单元，充分利用流水线。

程序优化：（7 分）

```
int i, j, k, r;
for(k = 0; k < n; k++)
{
    for(i = 0; i < n; i++)
    {
        r = A[i][k];
        for(j = 0; j < n - 1; j += 2){
            C[i][j] += r * B[k][j];
            C[i][j + 1] += r * B[k][j + 1];
        }
        for(; j < n; j++)
            C[i][j] += r * B[k][j];
    }
}
```

- 1) 循环展开，充分利用运算单元和流水线。
- 2) 改变循环次序，增加 cache 的命中率。