计 算 机 组 成 原 理 实 验 报 告

题目： 数据的存放顺序和对齐方式

姓　　名： 段欣然

专　　业： 计算机科学与技术

年　　级： 2020 级

学　　号： 202011081033

任课教师： 王志春

完成日期： 2022 年 5 月 6 日

人工智能学院

一、 实验要求

了解数据在机器中的存放方式。

二、 实验结果与分析

1. 设计一个程序以检查你的机器是大端方式还是小端方式。

Big endian 大端和 Little endian 小端的区别体现在对数据的存储和读取方式，大端方式将数据低位保存在内存的高位，小端模式则将数据高位保存在内存高位。因此只要知道储存数据的内存最低一位（或最高位）中保存的是否数据低位，就能判断是否为小端（或大端）存储。故代码如下，为 int 型数据赋值，注意要是该数每个 byte 存储的数据有区别，十六进制下该数每两位应不相同，例如 $0x0a0b0c0d$。然后考虑读取内存中最低一位 byte 数据，把该数据的存储位置赋给字符指针，使字符指针指向该数存储位置开端即低位。再将该指针对应字符转换为整型数据与原数高位和低位对比即可。

```cpp
#include<bits/stdc++.h>
using namespace std;
int main(){
    int a = 0x0a0b0c0d;
    // cout<<a<<endl;
    char* p = (char*)&a;
    // cout<<setbase(16)<<(int)*p<<endl;
    if((int)*p == 0x0d){
        cout<<"Big endian"<<endl;
    }
    else{
        cout<<"Little endian"<<endl;
    }
    return 0;
}
```

图表 1 判断大端、小端实现代码

结果如下

```
PS C:\Users\24636\AppData\Local\Temp> cd "
tempCodeRunnerFile.cpp -o tempCodeRunnerFi
Big endian
PS C:\Users\24636\AppData\Local\Temp>
```

图表 2 输出结果

故我的电脑是大端方式。

2. 设计一个程序以检查内存变量（如结构或数组）是否按边界对齐。

```cpp
#include<iostream>
using namespace std;
struct node{
    char c;
    int a;
    char b;
};
struct test{
    int a;
    char b,c;
};
int main(){
    cout<<"size of int: "<<sizeof(int)<<endl;
    cout<<"size of char: "<<sizeof(char)<<endl;
    cout<<"size of node: "<<sizeof(struct node)<<endl;
    cout<<"size of test: "<<sizeof(struct test)<<endl;
    return 0;
}
```

图表 3 定义结构体输出占用内存

```
size of test: 8
PS C:\Users\24636\AppData\Local\Temp> c
 ; if ($?) { g++ tempCodeRunnerFile.cp
\tempCodeRunnerFile }
size of int: 4
size of char: 1
size of node: 12
size of test: 8
PS C:\Users\24636\AppData\Local\Temp>
```

图表 4 运行结果

结构体 node 空间占用如下表（每格表示 1byte）

			b
a			
			c

结构体 test 空间占用如下表（每格表示 1byte）

		c	b
a			

故可以得知编译器自动对齐内存边界。

三、 实验小结

```
struct super{
    char c;
    double d;
    int s;
};
```

图表 5 结构体定义

定义如上结构体，下讨论该结构体占用内存为 24byte 而非 20byte。

先假设该结构体占用空间如下表，占用 20byte。

s			
d			
			c

考虑连续写入（或从内存中读取）两个上述结构体，有

s			
d			
			c
s			
d			
			c

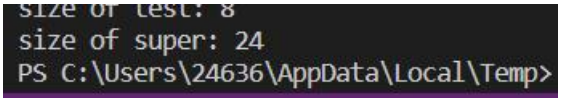
假设第一个结构体中 c 存储起始位置为a，d 的起始存储位置为$a+28$，无法被 8 整除，与边界对齐要求矛盾。

上述结构体正确的占用空间为：

s			
d			

			c

占用空间为 24byte。



图表 6 验证结果