Homework9

Homework9

- 题目1

```
struct data {
    char a;
    short b[2];
    char* c;
    union{
        char x;
        short y;
        int z;
    }p;
    char d;
};
struct data d[2];
```

- Suppose the address of global variable dis 0x8049600, please answer the following questions.

Homework9

- 题目1

Variable	Start address
d[0]	0x8049600
d[1]	[1]
d[0].a	[2]
d[0].b[1]	[3]
d[0].c	[4]
d[0].p.y	[5]
d[0].p.z	[6]
d[0].d	[7]

Homework9

- 题目2
  - What’s the output of the following C program? (on a 32-bit machine)

```
int main() {
    static char char_table[3][13] = {
            { 'd', 'o', 32, 'y', 'o', 'u', 32, 'w', 'a', 'n', 't', 32, 'a' }, { 32, 109, 105,
            100, 116, 101, 114, 109, 32, 101, 120, 97, 109 }, {0 }
    };
    static char ans[] = "abcdefghijklmnopqrstuvwxyyz";
    printf("%s?\n", char_table);
    printf("%c%c%c!\n",
            (char)(((char **)ans)[6]),
            (char)(((char *)ans)[4]),
            (char)(ans[18]));
    return 0;
}
```

Homework9

- 题目3
  - For each of the following structure declarations, determine the offset of each field, the total size of the structure, and its alignment requirement under x86-64.
  - A. struct P1 { int I; char c; long j; char d;};
  - B. struct P2 { long I; char c; char d; int j;};
  - C. struct P3 { short w[3]; char c*[3]};
  - D. struct P4 { struct P1 a[2]; struct P2 *p};
  - E. struct P5 { short w[3]; char c[3]}.

Homework9

- 题目3

	Offset of each field				Total size	Alignment
A	i:0	c:4				
B						
C						
D						
E						

Homework9

- 题目4
  - Suppose we have the following function ‘login’ to perform login process.

```
int login() {
    char username[8];
    char password[8];
    gets(username);
    gets(password);
    return check_match_in_database(username, password);
}
```

Homework9

- 题目4
  - Here is a part of the function’s assembly.

```
Pushl %ebp
movl %esp, %ebp
subl $40, %esp
leal -16(%ebp), %eax
movl %eax, (%esp)
call _gets
leal -24(%ebp), %eax
movl %eax, (%esp)
call _gets
……
```

Homework9

- 题目4
  - In the normal process, if the username and the password are both ok, the function ‘login_ok’ will be called to indicate login success. We’ve already known that the address of ‘login_ok’ is 0x804013da. Can you construct an input to make the function ‘login_ok’ be called after ‘login’ returns? You need to specify the key bytes and their positions rather than the complete input. And give one brief explanation about your input.