

```
char* ordersplit_result[10] = {0};
splitstr(g_buffer, ordersplit_result, "+=,\\r\\n");
```

```
int splitstr(const char* in, char** splitresult, const char* split_mark_table) {
    static char buf[300];
    strcpy(buf, in);
    int off = 0;

    *splitresult = buf;
    off++;

    for (size_t i = 0; buf[i] != '\\0'; i++) {
        if (strcontain2(buf[i], split_mark_table)) {
            buf[i] = '\\0';

            splitresult[off] = &buf[i+1];
            printf("%s\\r\\n", splitresult[off]);
            off++;
        }
    }

    return off - 1;
}
```

1 3.二重指针与指针数组

```
2
3 char buf[100];
4 char** spliterresult1;
5 char* ordersplit_result[10];
6 spliterresult1 = ordersplit_result;
7 spliterresult1[0] = ordersplit_result[0];
8 ordersplit_result[0] = "first";
9 ordersplit_result[1] = "second";
```

11 **spliterresult1是指向 ordersplit_result指针数组中的内容，*ordersplit_result是指向数组的内容的首地址，ordersplit_result是二重指针地址，*ordersplit_result是二重指针地址里面的值。

12 图中，将ordersplit_result的地址赋给splitresult，避免了野指针的出现，*spliterresult是buf的首地址，spliterresult[off]中存放的是每个分割完后下一个字符串开头的首地址。

