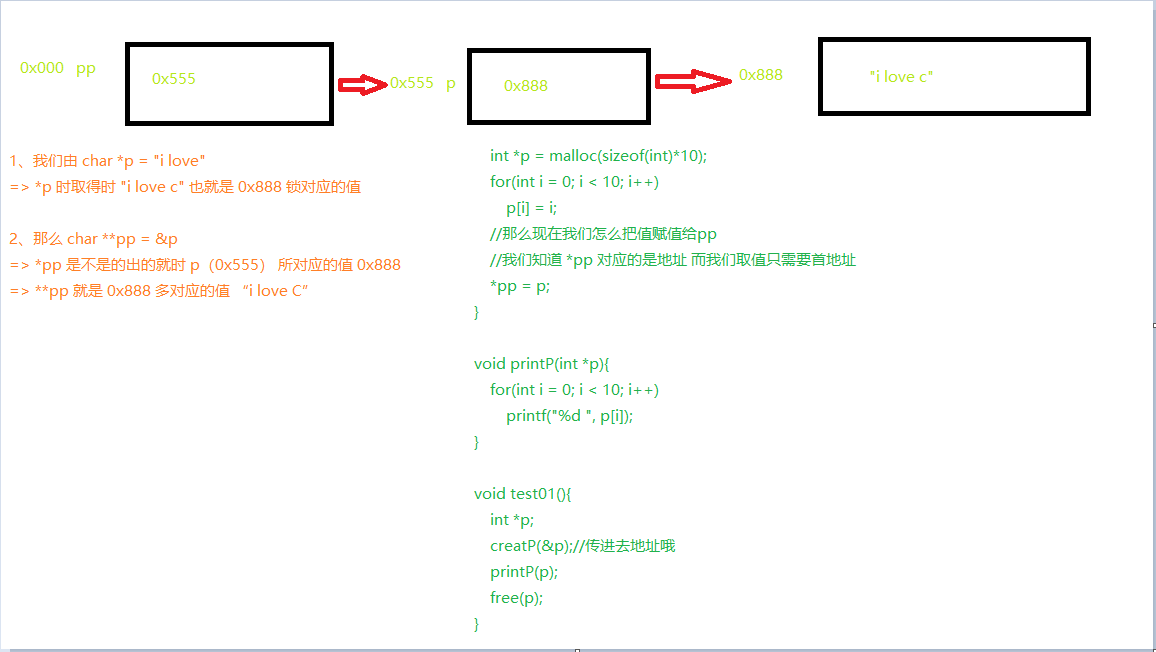
**二级指针详解**

1. 概念
   1. 不知道大家刚接触到指针有没有跟我一样，感到很懵逼，尤其是接触到二级指针、我的天呀，搞了好久没搞懂
   2. 现在我们用一张图来进行解释



1. 实战
   1. 那么现在我们用二级指针来操作一下文件的读写
   2. 代码

// 我们来获取一下当前文件的行数

int readLines(FILE \*fp){

int lines = 0;

char buff[1024] = {0}; // 要存放的下每一行的最大字符数

while(fgets(buff, sizeof(buff), fp) != NULL){

++lines;

memset(buff, 0, 1024); //把buff清空一下

}

//这边我们读完之后，我们的文件指针就跑到最后去，所以我们要再调回来

fseek(fp, 0, SEEK\_SET);

return lines;

}

void readContens(FILE \*fp, char \*\*fileContents, int lines){

if(lines <= 0){

printf("error\n");

exit(0);

}

if(fileContents == NULL){

printf("error");

exit(0);

}

int index = 0;

char buff[1024] = {0}; // 要存放的下每一行的最大字符数

while(fgets(buff, sizeof(buff), fp) != NULL){

//我们得到每一行的并存在buff中，那么我们是不是可以申请一个char \* 来存放

int lineLen = strlen(buff) + 1; //后面还有'\0'

char \*p = (char \*)malloc(sizeof(char)\*lineLen);

strcpy(p, buff);

fileContents[index++] = p; //把p的值放到fileContents上面去

memset(buff, 0, 1024); //把buff清空一下

}

fseek(fp, 0, SEEK\_SET);

}

void printContent(char \*\*content, int lines){

for(int i = 0; i < lines; i++){

printf("%s", content[i]);

}

}

/\*\* 释放内存空间

\*

\* 我们开辟了一个 char \*\*

\* 还有一个 char \*

\* 那我们应该先for遍历释放掉char \*

\* 再释放掉char \*\* 如果次操作相反，是不是 char \* 就没办法释放了

\*/

void freeSpace(char \*\*p, int lines){

for(int i = 0; i < lines; i++)

free(p[i]);

free(p);

p = NULL;

}

void test(){

FILE \*fp;

fp = fopen("test.txt", "r");

//获取文件的行数

int lines = readLines(fp);

printf("lines = %d\n", lines);

//获取文件的内容

char \*\*fileContents = (char \*\*)malloc(sizeof(char \*)\*lines);

readContens(fp, fileContents, lines); //那么现在我们把值放在fileContents中去了

printContent(fileContents, lines);

// 获取完，关闭文件

fclose(fp);

//现在删除掉内存空间

freeSpace(fileContents, lines);

if(fileContents == NULL)

printf("free succeed");

}