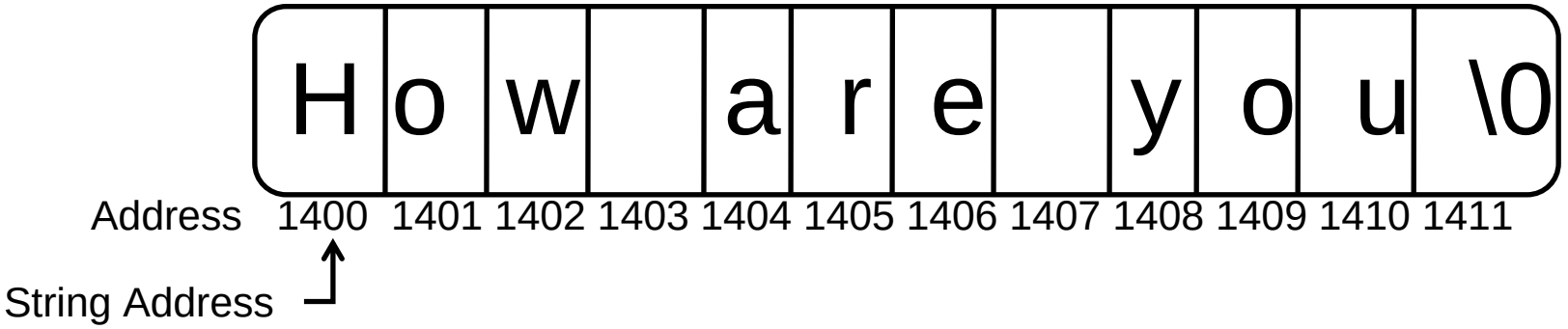
11

Case Study of Pointers (part 2)

Case 1: Copy String Using Its Pointer

- `Problem definition`
  - **All the variables have its own address, including the strings. Then, we can use the pointer to copy a string to the other. We can also just copy a piece of string by using the pointer.**



Source Code

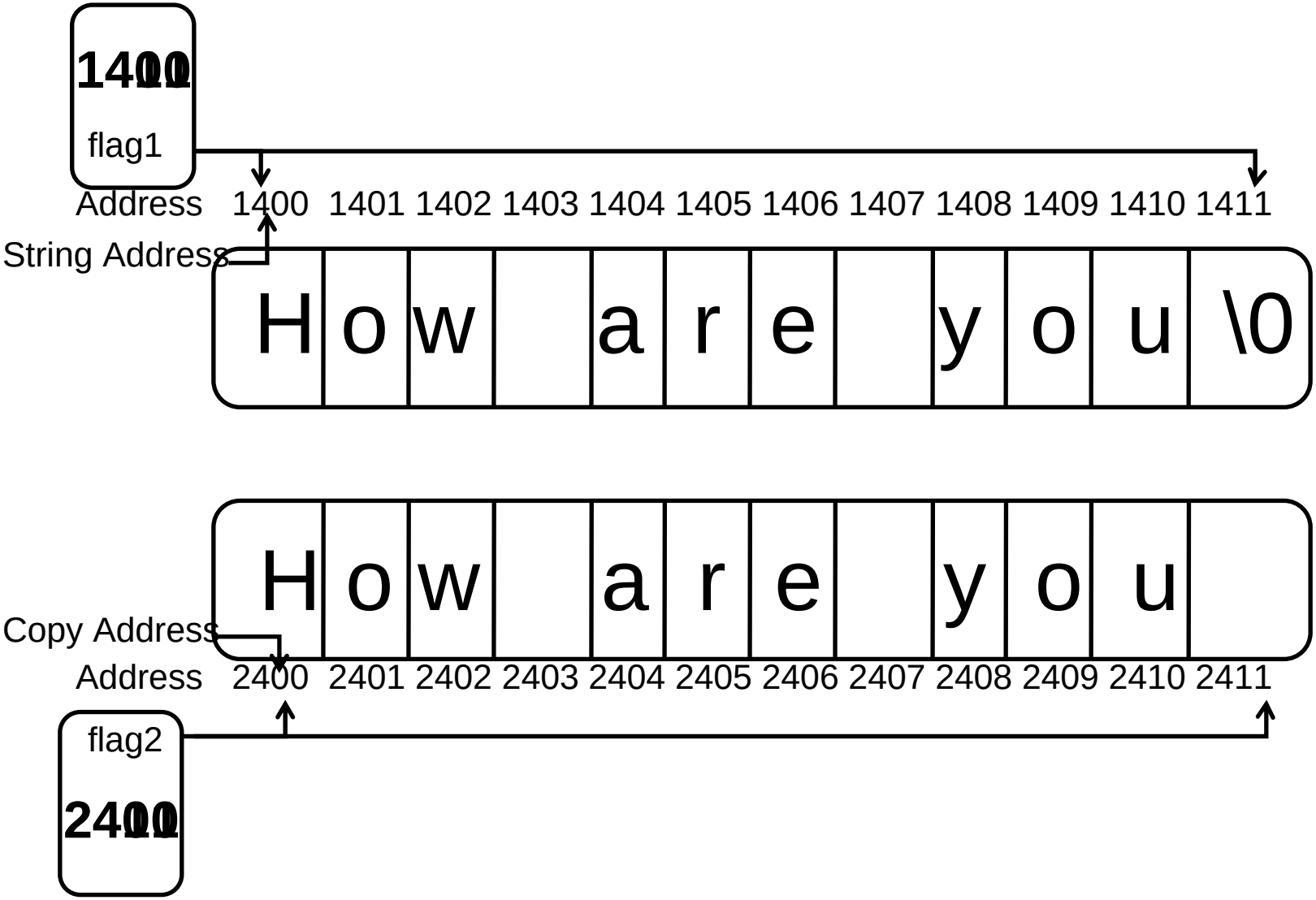
```
#include <stdio.h>
#include <stdlib.h>

int main(void){
    char string[1001];
    char copy[1001];
    char *flag1, *flag2;

    printf("請輸入要複製的字串(1000字元內): ");
    gets(string);

//將指標設定到要被複製的字串與複製過後的字串
    flag1=string;
    flag2=copy;

//進行字串複製
    while(*flag1!='\0'){
                        *flag2=*flag1;
                        flag1++;
                        flag2++;
    }
    *flag2='\0';

    printf("複製出的新字串： ");
    puts(copy);

    system("pause");
    return 0;
}
```

宣告字串最多只能存1000個，第1001個為放'\0'的字元

並且宣告Flag1、Flag2兩個字元指標

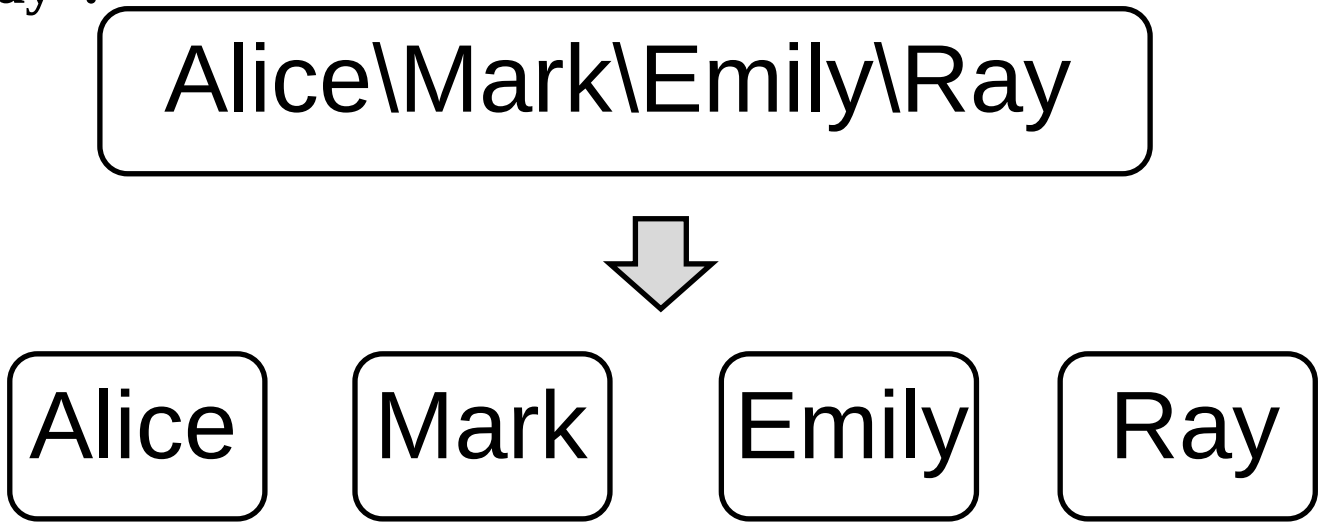
```
/* 當指標掃到*flag1為字串結尾時，停止複製 */
/* 將*flag1的內容複製到*flag2 */
/* flag1與flag2都往前進一格 */


/*在複製後的字串最後補上結尾'\0' */
```

Illustration



Case 2: Cut a String into Several Small Ones

- `Problem definition`
  - **Cut a string into several small strings means if there is a long string, we can use pointer to cut it into pieces by some special symbol.**
  - **For a example, there is a string "Alice\Mark\Emily\Ray" which is separated by "\". By running this program, we will get four new strings : "Alice" "Mark" "Emily" "Ray".**



Source Code (1/2)

```
#include <stdio.h>
#include <stdlib.h>

int main(void){
    char string1[1001];
    char *flag1, *flag2;
    char *ptr[10];
    int i=0,j=0;
    char separate;

// 取得欲分隔的字串
    printf("請輸入拿來當分隔的單一字元: ");
    separate=getche();
    printf("\n");

    printf("請輸入小於1000字元的字串，並以剛剛選定的字元分隔： " );
    gets(string1);
    printf("\n");

    flag1=string1;
    flag2=flag1;
```

宣告字串最多只能存1000個，第1001個為放'\0'的字元

並且宣告Flag1、Flag2兩個字元指標

字串指標陣列ptr用來存放分隔過的字串

i與j用來標示目前指定到的陣列區塊

Source Code (2/2)

分隔字串的程式主要區塊

```
//判斷是否需要分隔
    do{
                if(*flag1==separate){
                                *flag1='\0';
                                ptr[i]=flag2;
                                flag2=flag1+1;
                                i++;
                }
                flag1++;
    }while(*flag1!='\0');

    ptr[i]=flag2;

    printf("分隔後的string為： \n" );

//印出分隔以後的字串
    for(j=0;j<=i;j++){
                        puts(ptr[j]);
    }

    system("pause");
    return 0;
}
```

```
/*若指標所指到的為被定義是分隔的符號的話*/
/*將分隔的符號以'/0'取代，代表是一小段字串的終結*/
/*將被分隔的小段字串存到另一個區塊中*/
/*flag2移到下一個小字串的開頭*/
/*儲存區塊移到下一個*/

/*掃描下一個位元*/
/*若flag1直接就找到'/0'，則跳出迴圈*/

/*補上最後一段小字串*/
```

Illustration