

```
psc@ubuntu: ~  
psc@ubuntu:~$ ls  
2020203002 Desktop execve malloc.c pause signal.c Templates  
alarm.c Documents execve.c malloc.out pause.c sigprocmask vector.cpp  
alarm.c Downloads execvp mecro1 Pictures sigprocmask.c vector.out  
array1.cpp execl.c execvp.c mecro1.c Public sigset_t Videos  
array1.out execl.c fork mecro2 sigaction sigset_t.c wait  
array.cpp execv fork.c mecro2.c sigaction.c string.cpp wait.c  
array.out execv.c main.sh Music signal string.out  
psc@ubuntu:~$ cat malloc.c  
#include <stdio.h>  
#include <stdlib.h>  
  
#define SAFE_FREE(p) if (p) { free(p); p = NULL; }  
#define MAX_ITEM_CNT 5  
  
int main(void){  
    int i = 0;  
  
    int * buf1 = (int*)malloc(MAX_ITEM_CNT * sizeof(int));  
    int * buf2 = (int*)calloc(MAX_ITEM_CNT, sizeof(int));  
  
    printf("Address of buf1 : %p\n", buf1);  
    for (i = 0; i < MAX_ITEM_CNT; i++) { printf("%d\n", *(buf1 + i)); }  
  
    printf("Address of buf2 : %p\n", buf2);  
    for(i = 0; i < MAX_ITEM_CNT; i++) { printf("%d\n", *(buf2 + i)); }  
  
    SAFE_FREE(buf1);  
    SAFE_FREE(buf2);  
  
    return 0;  
}  
psc@ubuntu:~$ ./malloc.out  
Address of buf1 : 0x55c01b4342a0  
0  
0  
0  
0  
0  
Address of buf2 : 0x55c01b4342c0  
0  
0  
0  
0  
0  
psc@ubuntu:~$
```

```
psc@ubuntu: ~  
psc@ubuntu:~$ ls  
2020203002 Documents execvp mecro1.c realloc1.c sigprocmask vector.out  
alarm Downloads execvp.c mecro2 realloc1.out sigprocmask.c Videos  
alarm.c execl fork mecro2.c realloc2.c sigset_t wait  
array1.cpp execl.c fork.c Music realloc2.out sigset_t.c wait.c  
array1.out execv main.sh pause sigaction string.cpp  
array.cpp execv.c malloc.c pause.c sigaction.c string.out  
array.out execve malloc.out Pictures signal Templates  
Desktop execve.c mecro1 Public signal.c vector.cpp  
psc@ubuntu:~$ cat realloc1.c  
#include <stdio.h>  
#include <stdlib.h>  
  
#define SAFE_FREE(p) if (p) { free(p); p = NULL; }  
#define MAX_BUF_SIZE 16  
  
int main(void){  
    char * buf = (char *)malloc(MAX_BUF_SIZE * sizeof(char));  
    printf("Address of buf : %p\n", buf);  
  
    buf = (char *)realloc(buf, 2 * MAX_BUF_SIZE * sizeof(char));  
    printf("Address of buf : %p\n", buf);  
  
    SAFE_FREE(buf);  
  
    return 0;  
}  
psc@ubuntu:~$ ./realloc1.out  
Address of buf : 0x55e1e0e122a0  
Address of buf : 0x55e1e0e126d0  
psc@ubuntu:~$
```

```
psc@ubuntu: ~  
psc@ubuntu:~$ ls  
2020203002 Documents execvp mecro1.c realloc1.c sigprocmask vector.out  
alarm Downloads execvp.c mecro2 realloc1.out sigprocmask.c Videos  
alarm.c execl fork mecro2.c realloc2.c sigset_t wait  
array1.cpp execl.c fork.c Music realloc2.out sigset_t.c wait.c  
array1.out execv main.sh pause sigaction string.cpp  
array.cpp execv.c malloc.c pause.c sigaction.c string.out  
array.out execve malloc.out Pictures signal Templates  
Desktop execve.c mecro1 Public signal.c vector.cpp  
psc@ubuntu:~$ cat realloc2.c  
#include <stdio.h>  
#include <stdlib.h>  
  
#define SAFE_FREE(p) if (p) { free(p); p = NULL; }  
#define MAX_BUF_SIZE 16  
  
int main(void){  
    char * buf1 = (char *)malloc(MAX_BUF_SIZE * sizeof(char));  
    char * buf2 = (char *)malloc(MAX_BUF_SIZE * sizeof(char));  
  
    printf("Address of buf1 : %p\n", buf1);  
  
    buf1 = (char *)realloc(buf1, 2 * MAX_BUF_SIZE * sizeof(char));  
    printf("Address of buf1 : %p\n", buf1);  
  
    SAFE_FREE(buf1);  
    SAFE_FREE(buf2);  
  
    return 0;  
}  
psc@ubuntu:~$ ./realloc2.out  
Address of buf1 : 0x558e9c2552a0  
Address of buf1 : 0x558e9c2556f0  
psc@ubuntu:~$
```