Overlapping chunks

2019-07-11

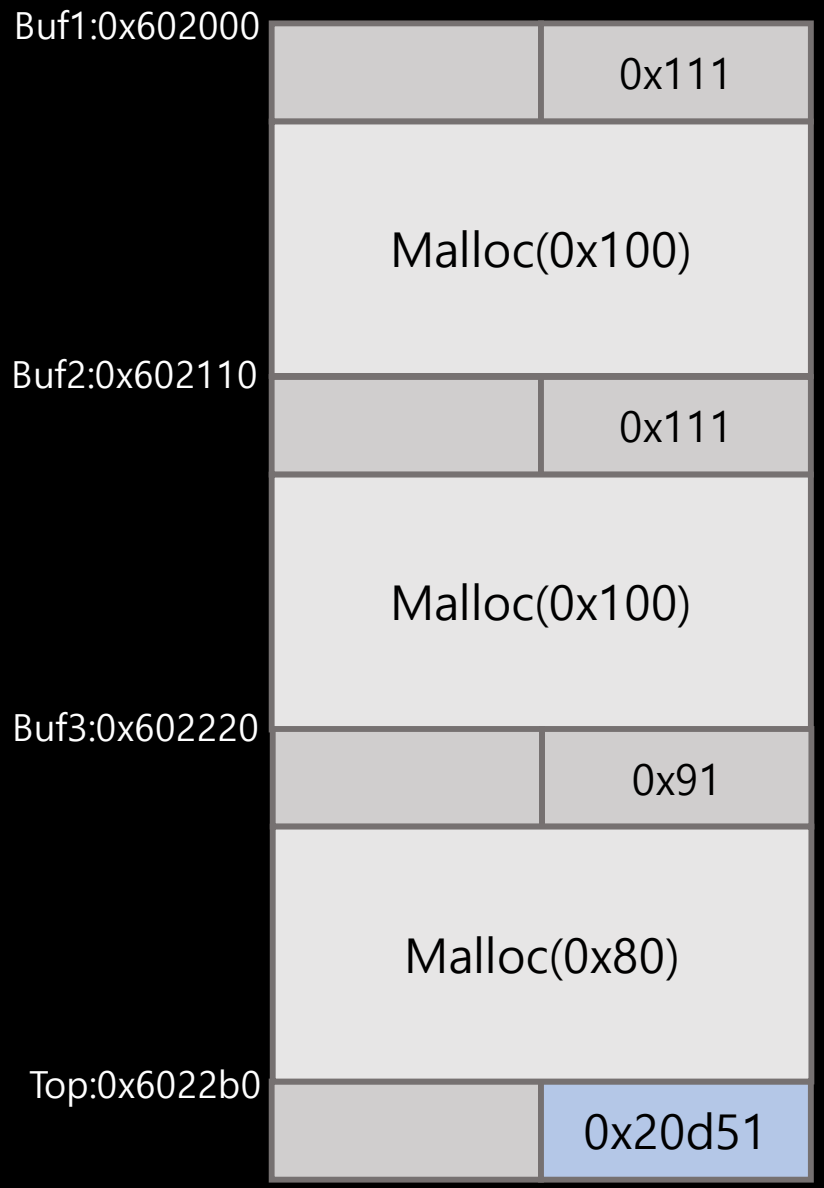
(쉬다온) 서동훈

Source code

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdint.h>

void main(){

    char *buf1 = malloc(0x100);
    char *buf2 = malloc(0x100);
    char *buf3 = malloc(0x80);

    memset(buf1, 'A', 0x100);
    memset(buf2, 'B', 0x100);
    memset(buf3, 'C', 0x80);

    free(buf2);

    int size;

    scanf("%272s",buf1);
    scanf("%d",&size);

    char *buf4 = malloc(size);

    scanf("%384s",buf4);
    printf("buf3 : %s\n",buf3);

    scanf("%128s",buf3);
    printf("buf4 : %s\n", buf4);
}
```

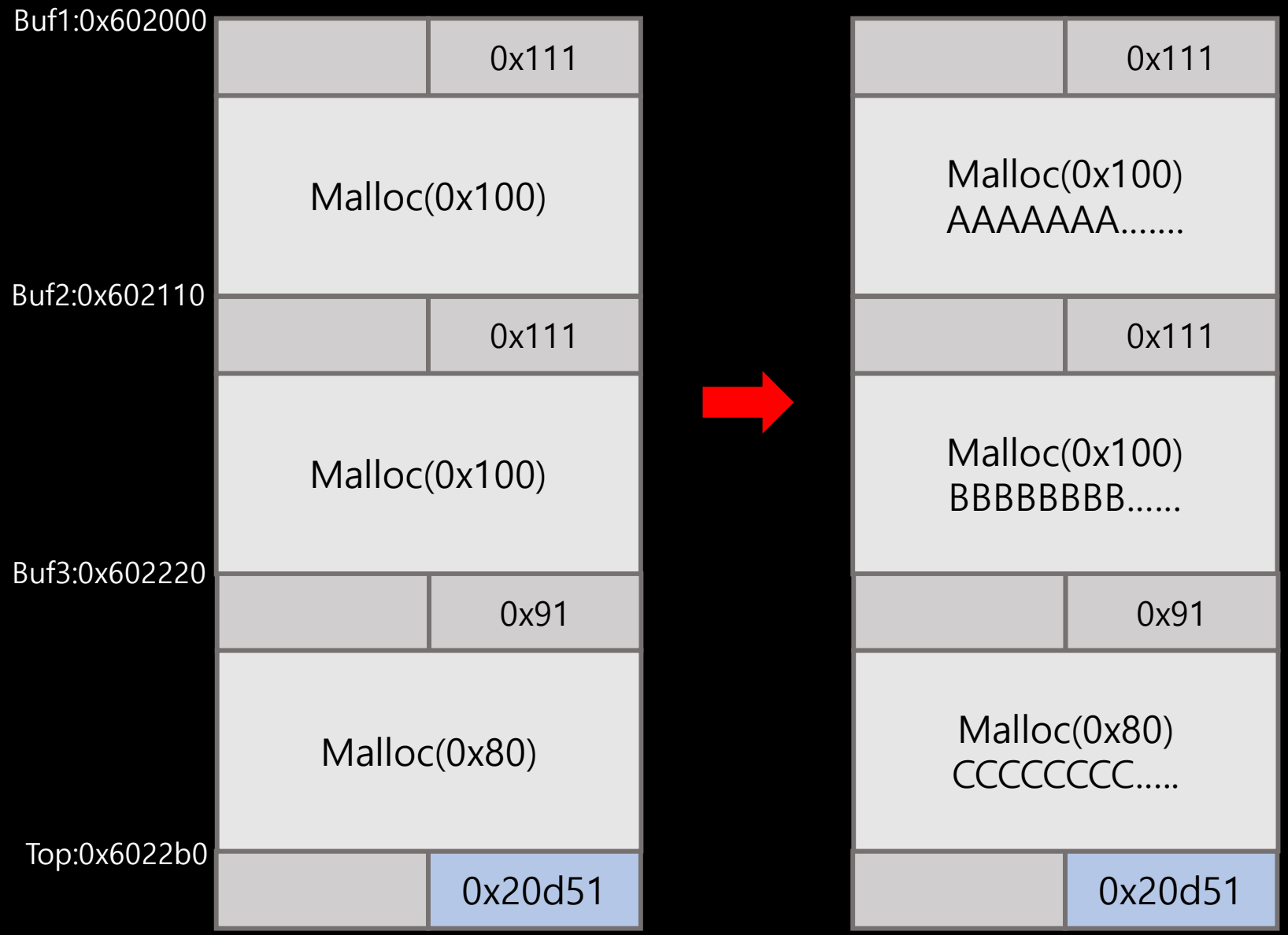
malloc * 3

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdint.h>

void main(){

    char *buf1 = malloc(0x100);
    char *buf2 = malloc(0x100);
    char *buf3 = malloc(0x80);

    memset(buf1, 'A', 0x100);
    memset(buf2, 'B', 0x100);
    memset(buf3, 'C', 0x80);

    free(buf2);

    int size;

    scanf("%272s",buf1);
    scanf("%d",&size);

    char *buf4 = malloc(size);

    scanf("%384s",buf4);
    printf("buf3 : %s\n",buf3);

    scanf("%128s",buf3);
    printf("buf4 : %s\n", buf4);
}
```



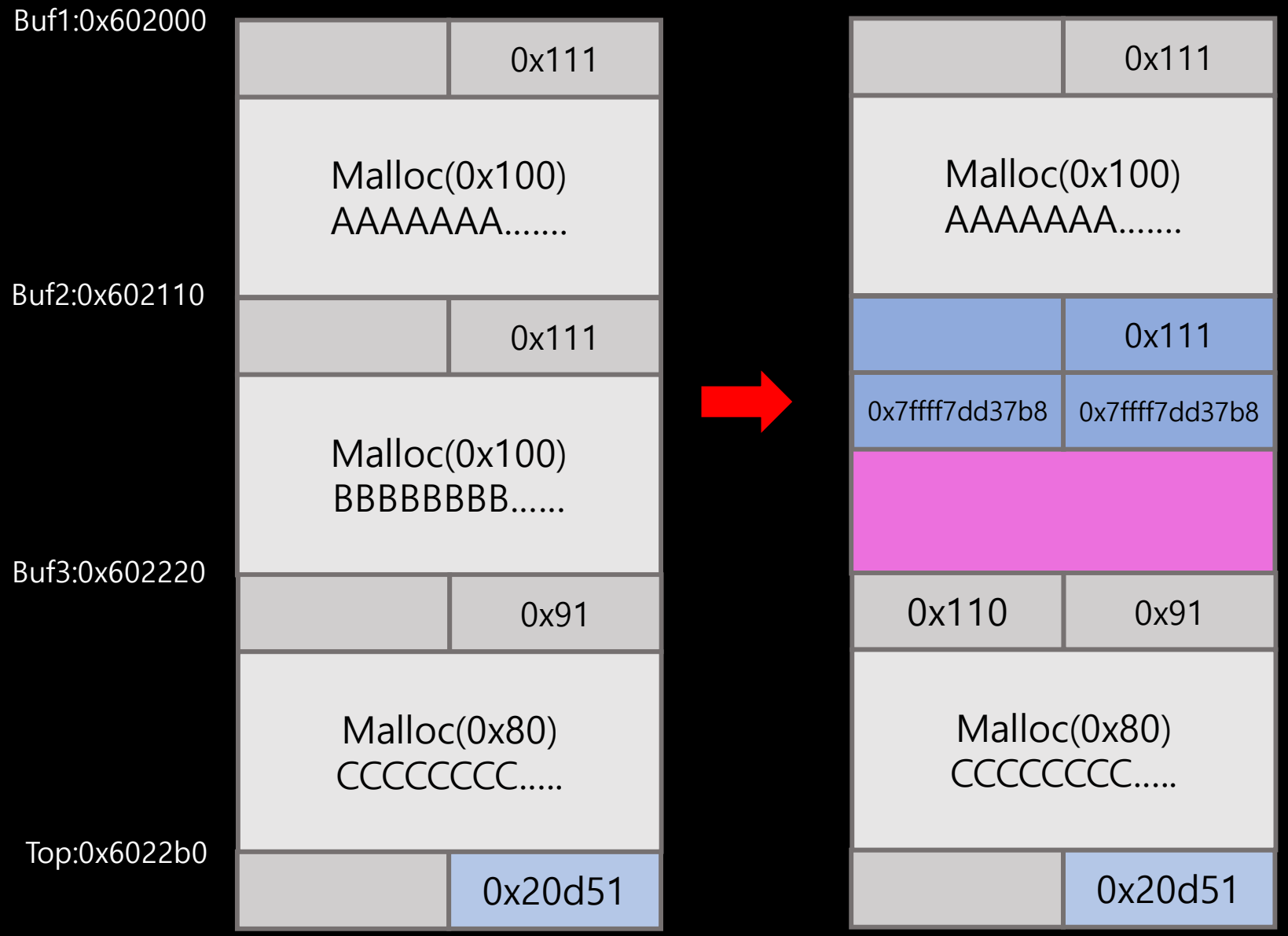
memset

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdint.h>

void main(){

    char *buf1 = malloc(0x100);
    char *buf2 = malloc(0x100);
    char *buf3 = malloc(0x80);

    memset(buf1, 'A', 0x100);
    memset(buf2, 'B', 0x100);
    memset(buf3, 'C', 0x80);

    free(buf2);

    int size;

    scanf("%272s",buf1);
    scanf("%d",&size);

    char *buf4 = malloc(size);

    scanf("%384s",buf4);
    printf("buf3 : %s\n",buf3);

    scanf("%128s",buf3);
    printf("buf4 : %s\n", buf4);
}
```



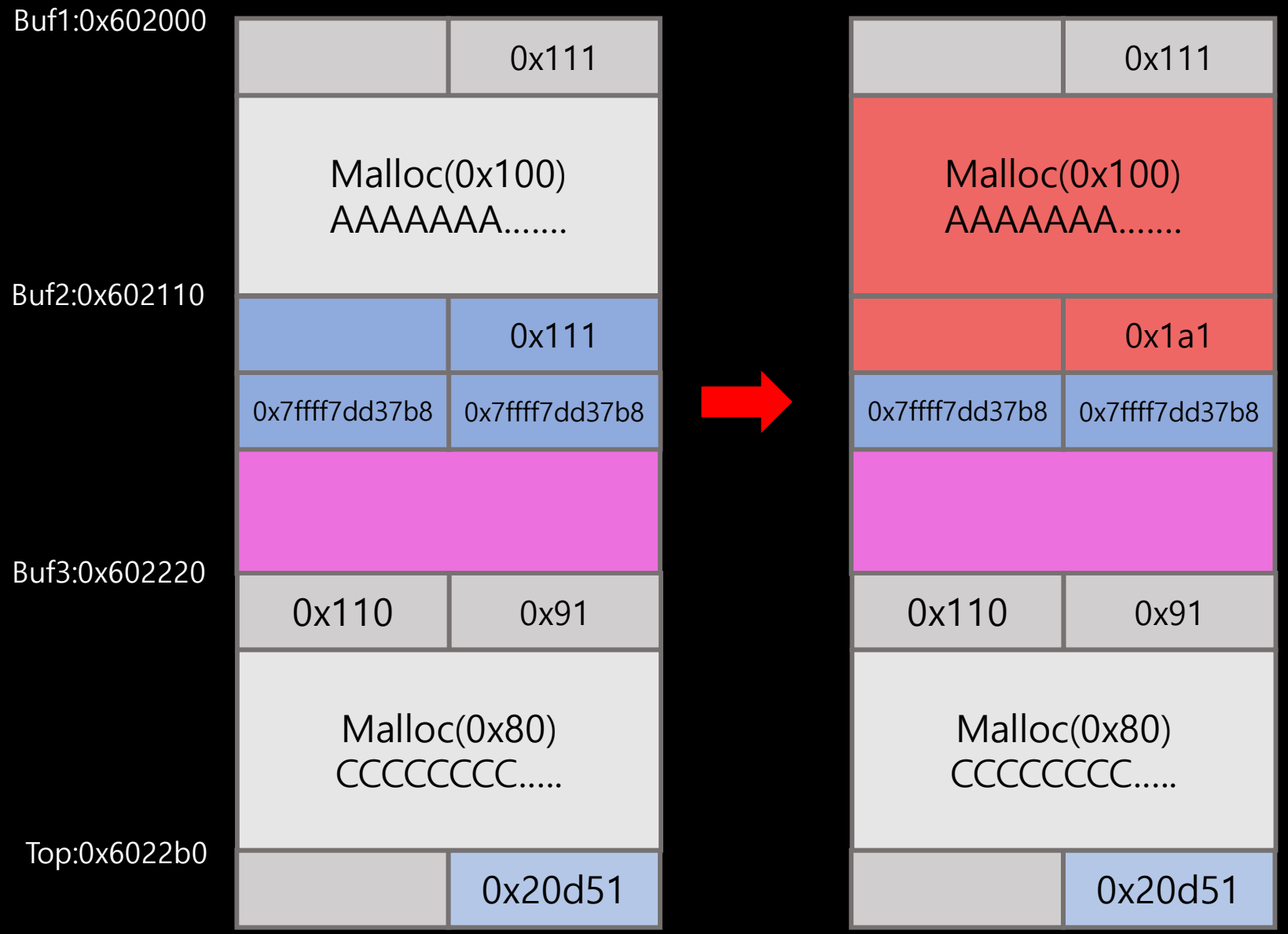
Free(buf2)

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdint.h>

void main(){

    char *buf1 = malloc(0x100);
    char *buf2 = malloc(0x100);
    char *buf3 = malloc(0x80);

    memset(buf1, 'A', 0x100);
    memset(buf2, 'B', 0x100);
    memset(buf3, 'C', 0x80);

    free(buf2);

    int size;

    scanf("%272s",buf1);
    scanf("%d",&size);

    char *buf4 = malloc(size);

    scanf("%384s",buf4);
    printf("buf3 : %s\n",buf3);

    scanf("%128s",buf3);
    printf("buf4 : %s\n", buf4);
}
```



Over write

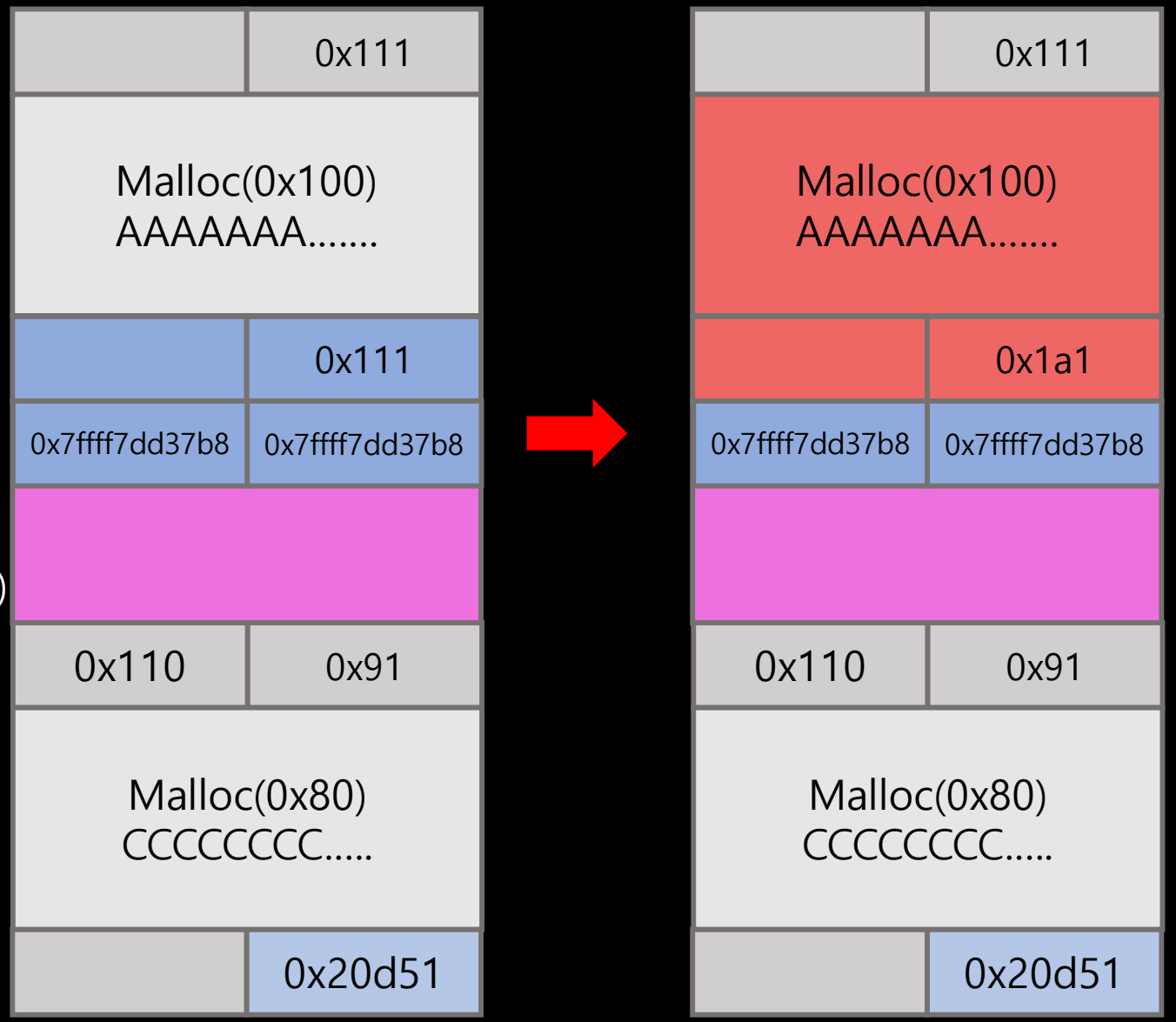
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdint.h>

void main(){

    char *buf1 = malloc(0x100);
    char *buf2 = malloc(0x100);
    char *buf3 = malloc(0x80);

    memset(buf1, 'A', 0x100);
    memset(buf2, 'B', 0x100);
    memset(buf3, 'C', 0x80);

    free(buf2);

    int size;

    scanf("%272s",buf1);
    scanf("%d",&size);

    char *buf4 = malloc(size);

    scanf("%384s",buf4);
    printf("buf3 : %s\n",buf3);

    scanf("%128s",buf3);
    printf("buf4 : %s\n", buf4);
}
```



Input size

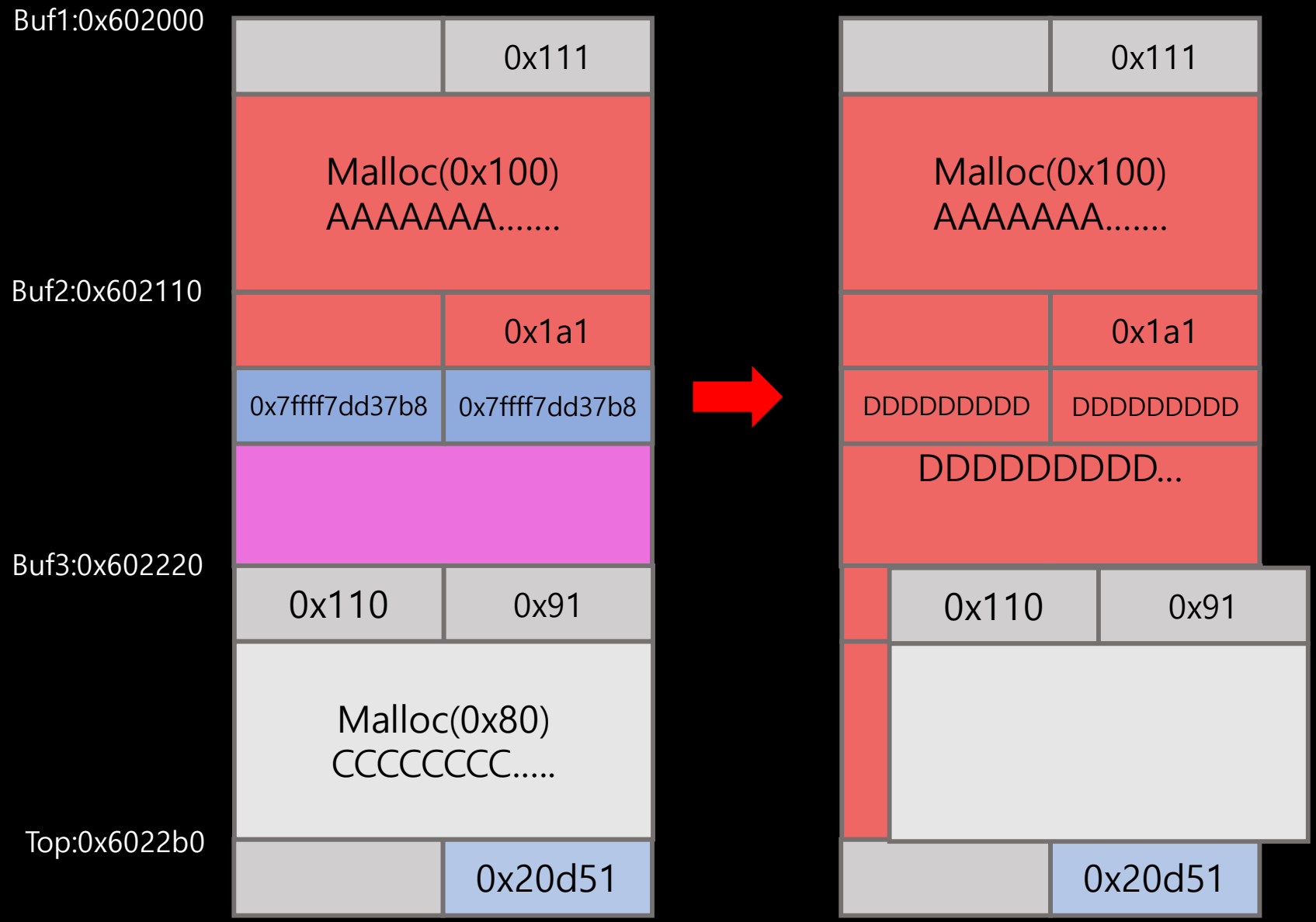
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdint.h>

void main(){

    char *buf1 = malloc(0x100);
    char *buf2 = malloc(0x100);
    char *buf3 = malloc(0x80);

    memset(buf1, 'A', 0x100);
    memset(buf2, 'B', 0x100);
    memset(buf3, 'C', 0x80);

    free(buf2);

    int size;

    scanf("%272s",buf1);
    scanf("%d",&size);

    char *buf4 = malloc(size);

    scanf("%384s",buf4);
    printf("buf3 : %s\n",buf3);

    scanf("%128s",buf3);
    printf("buf4 : %s\n", buf4);
}
```

Buf2(size) + Buf3(size)
0x110 + 0x90 = 0x1a0
0x1a0 - 0x10 = 0x190(400)



Overlapping chunks

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdint.h>

void main(){

    char *buf1 = malloc(0x100);
    char *buf2 = malloc(0x100);
    char *buf3 = malloc(0x80);

    memset(buf1, 'A', 0x100);
    memset(buf2, 'B', 0x100);
    memset(buf3, 'C', 0x80);

    free(buf2);

    int size;

    scanf("%272s",buf1);
    scanf("%d",&size);

    char *buf4 = malloc(size);

    scanf("%384s",buf4);
    printf("buf3 : %s\n",buf3);

    scanf("%128s",buf3);
    printf("buf4 : %s\n", buf4);
}
```



실습 환경

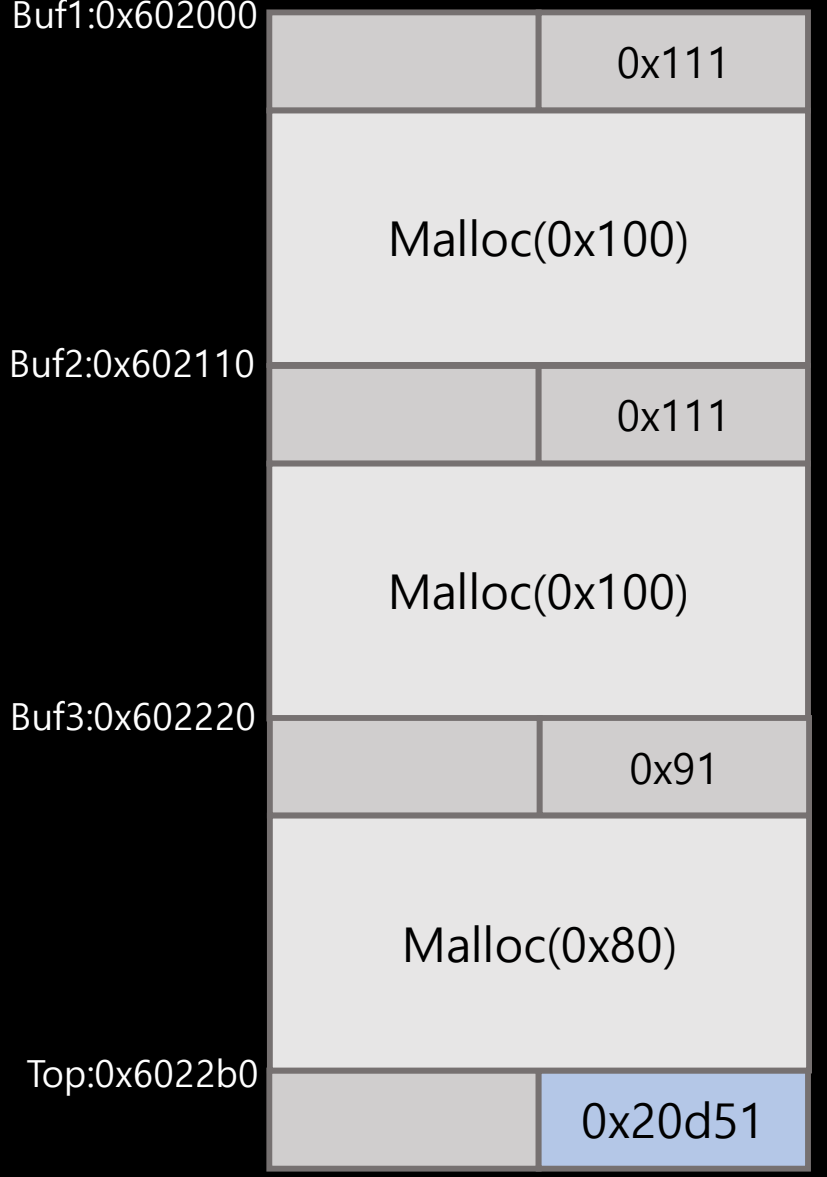
14.04.5 LTS

Glibc 2.19

64bit

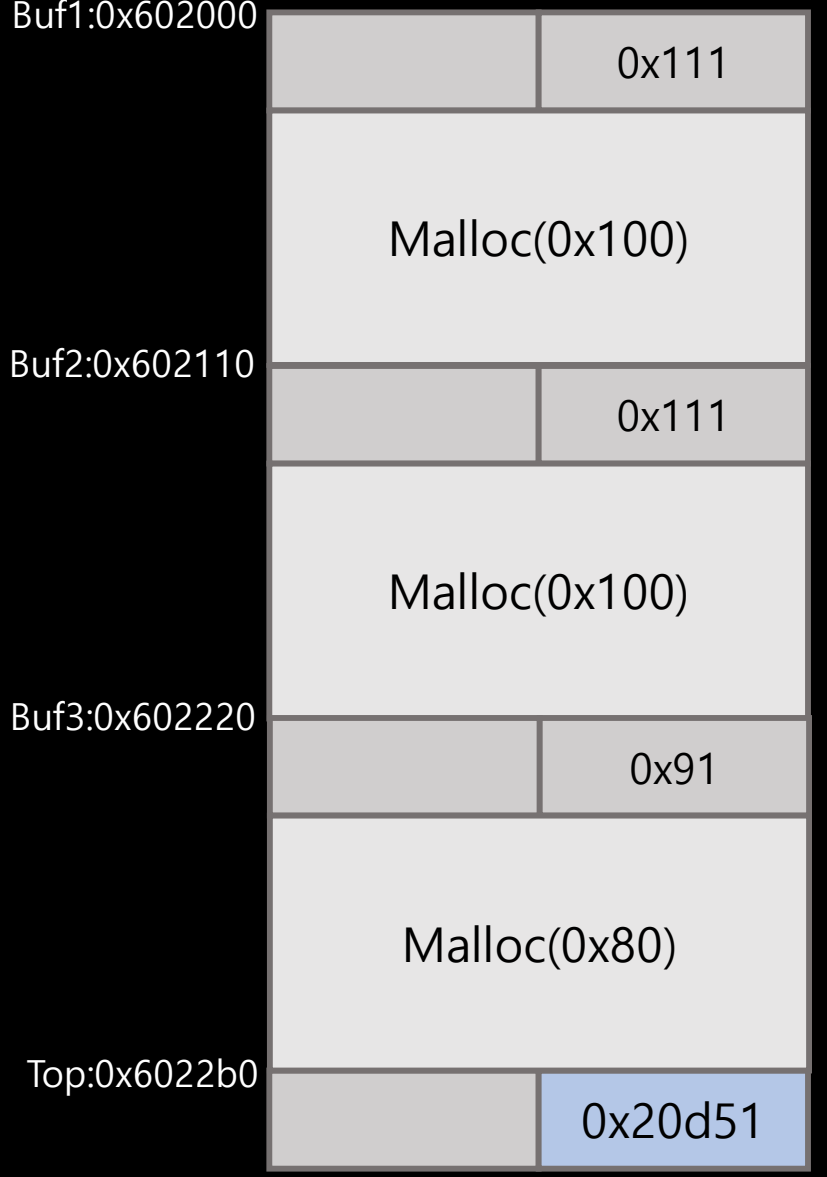
malloc * 3

```
gdb-peda$ x/24gx 0x00602000
0x602000:       0x0000000000000000      0x0000000000000111
0x602010:       0x0000000000000000      0x0000000000000000
0x602020:       0x0000000000000000      0x0000000000000000
0x602030:       0x0000000000000000      0x0000000000000000
0x602040:       0x0000000000000000      0x0000000000000000
0x602050:       0x0000000000000000      0x0000000000000000
0x602060:       0x0000000000000000      0x0000000000000000
0x602070:       0x0000000000000000      0x0000000000000000
0x602080:       0x0000000000000000      0x0000000000000000
0x602090:       0x0000000000000000      0x0000000000000000
0x6020a0:       0x0000000000000000      0x0000000000000000
0x6020b0:       0x0000000000000000      0x0000000000000000
gdb-peda$ x/24gx 0x00602110
0x602110:       0x0000000000000000      0x0000000000000111
0x602120:       0x0000000000000000      0x0000000000000000
0x602130:       0x0000000000000000      0x0000000000000000
0x602140:       0x0000000000000000      0x0000000000000000
0x602150:       0x0000000000000000      0x0000000000000000
0x602160:       0x0000000000000000      0x0000000000000000
0x602170:       0x0000000000000000      0x0000000000000000
0x602180:       0x0000000000000000      0x0000000000000000
0x602190:       0x0000000000000000      0x0000000000000000
0x6021a0:       0x0000000000000000      0x0000000000000000
0x6021b0:       0x0000000000000000      0x0000000000000000
0x6021c0:       0x0000000000000000      0x0000000000000000
```



malloc * 3

```
gdb-peda$ x/24gx 0x00602000
0x602000:       0x0000000000000000      0x0000000000000111
0x602010:       0x0000000000000000      0x0000000000000000
0x602020:       0x0000000000000000      0x0000000000000000
0x602030:       0x0000000000000000      0x0000000000000000
0x602040:       0x0000000000000000      0x0000000000000000
0x602050:       0x0000000000000000      0x0000000000000000
0x602060:       0x0000000000000000      0x0000000000000000
0x602070:       0x0000000000000000      0x0000000000000000
0x602080:       0x0000000000000000      0x0000000000000000
0x602090:       0x0000000000000000      0x0000000000000000
0x6020a0:       0x0000000000000000      0x0000000000000000
0x6020b0:       0x0000000000000000      0x0000000000000000
gdb-peda$ x/24gx 0x00602110
0x602110:       0x0000000000000000      0x0000000000000111
0x602120:       0x0000000000000000      0x0000000000000000
0x602130:       0x0000000000000000      0x0000000000000000
0x602140:       0x0000000000000000      0x0000000000000000
0x602150:       0x0000000000000000      0x0000000000000000
0x602160:       0x0000000000000000      0x0000000000000000
gdb-peda$ x/24gx 0x00602220
0x602220:       0x0000000000000000      0x0000000000000091
0x602230:       0x0000000000000000      0x0000000000000000
0x602240:       0x0000000000000000      0x0000000000000000
0x602250:       0x0000000000000000      0x0000000000000000
0x602260:       0x0000000000000000      0x0000000000000000
0x602270:       0x0000000000000000      0x0000000000000000
0x602280:       0x0000000000000000      0x0000000000000000
0x602290:       0x0000000000000000      0x0000000000000000
0x6022a0:       0x0000000000000000      0x0000000000000000
0x6022b0:       0x0000000000000000      0x0000000000020d51
```



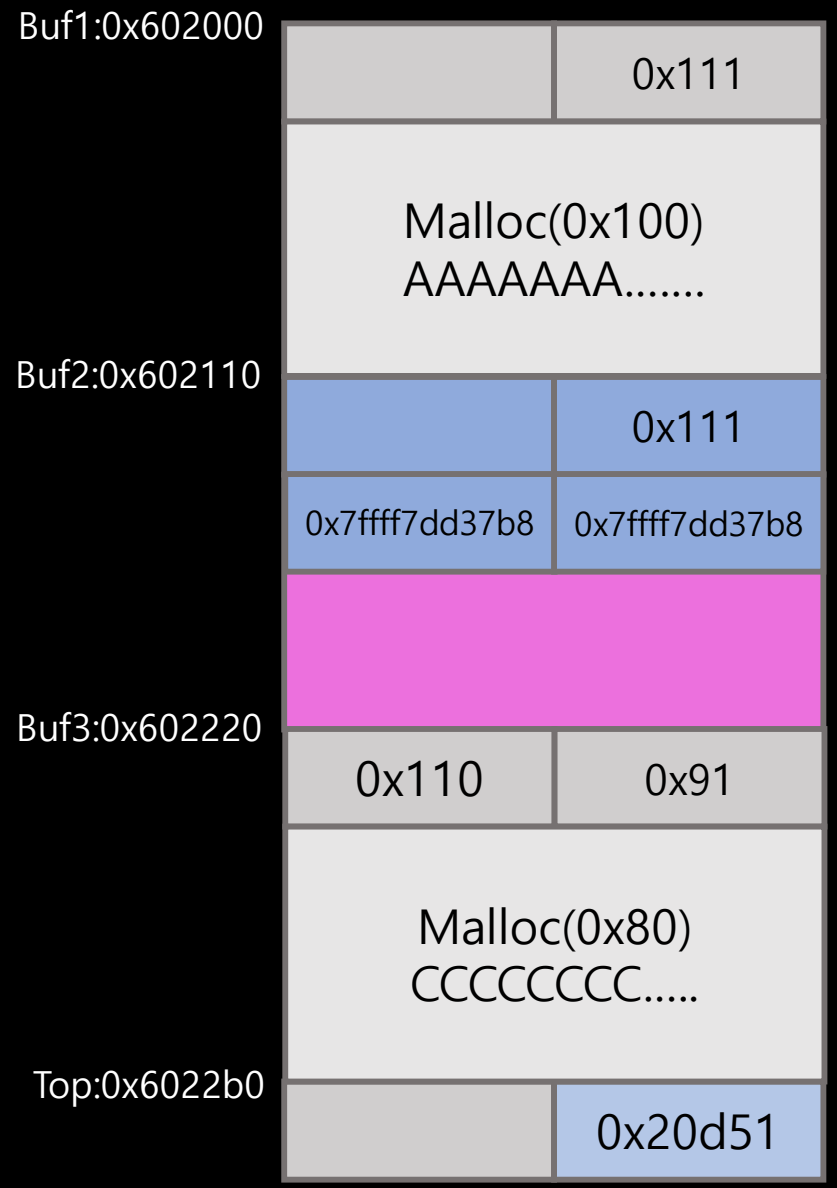
Free(buf2)

```
gdb-peda$ x/36gx 0x00602110
0x602110:       0x0000000000000000      0x0000000000000111
0x602120:       0x00007ffff7dd37b8      0x00007ffff7dd37b8
0x602130:       0x4242424242424242      0x4242424242424242
```

```
gdb-peda$ p main_arena
$6 = {
  mutex = 0x0,
  flags = 0x1,
  fastbinsY = {0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0},
  top = 0x602450,
  last_remainder = 0x0,
  bins = {0x7ffff7dd37b8, 0x7ffff7dd37b8, 0x7ffff7dd37c8, 0x7ffff7dd37c8,
    0x7ffff7dd37d8, 0x7ffff7dd37d8, 0x7ffff7dd37e8, 0x7ffff7dd37e8,
    0x7ffff7dd37f8, 0x7ffff7dd37f8, 0x7ffff7dd3808, 0x7ffff7dd3808,
    0x7ffff7dd3818, 0x7ffff7dd3818, 0x7ffff7dd3828, 0x7ffff7dd3828,
    0x7ffff7dd3838, 0x7ffff7dd3838, 0x7ffff7dd3848, 0x7ffff7dd3848,
    0x7ffff7dd3858, 0x7ffff7dd3858, 0x7ffff7dd3868, 0x7ffff7dd3868,
```

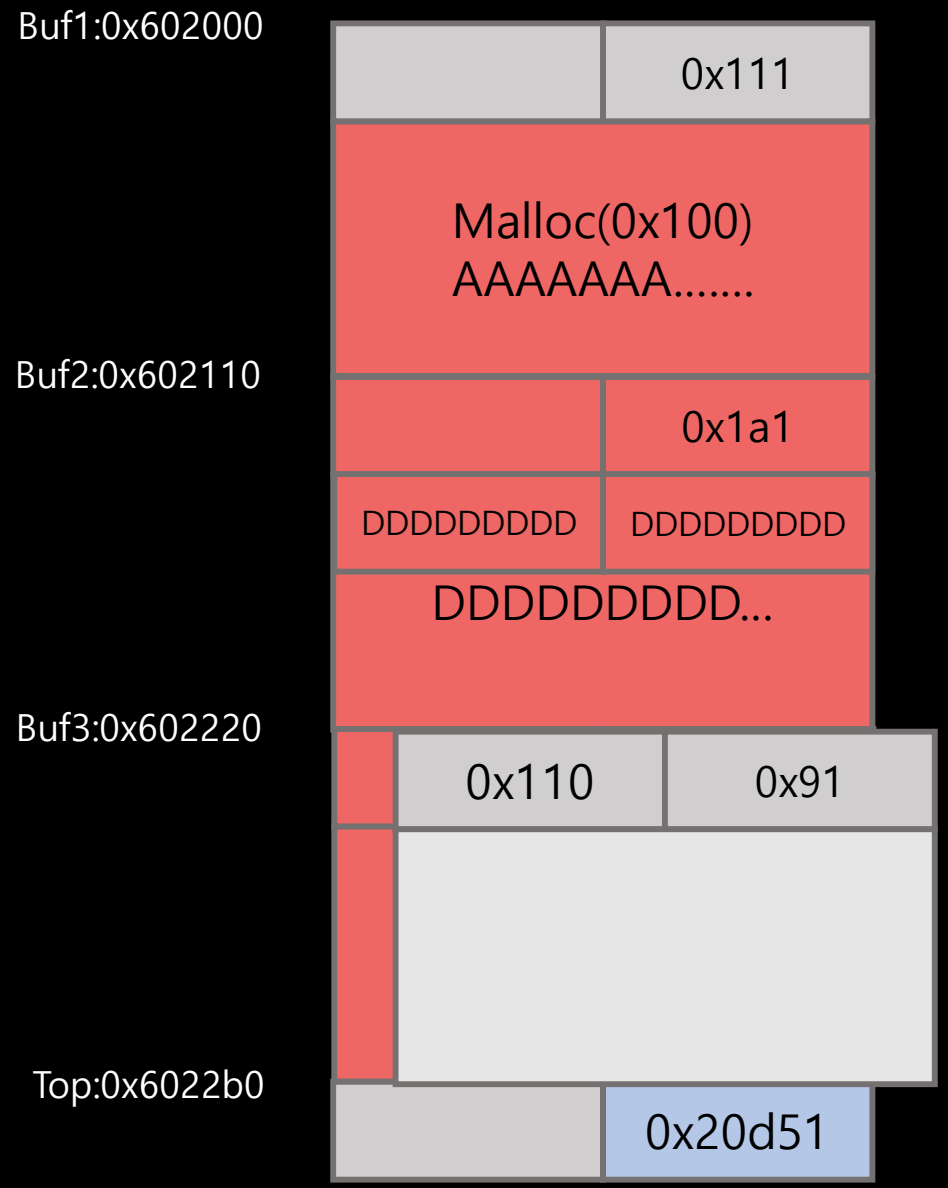
```
0x602220:       0x0000000000000110      0x0000000000000090
```

```
gdb-peda$ p main_arena
$3 = {
  mutex = 0x0,
  flags = 0x1,
  fastbinsY = {0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0},
  top = 0x6022b0,
  last_remainder = 0x0,
  bins = {0x602110, 0x602110, 0x7ffff7dd37c8, 0x7ffff7dd37c8, 0x7ffff7dd37d8,
    0x7ffff7dd37d8, 0x7ffff7dd37e8, 0x7ffff7dd37e8, 0x7ffff7dd37f8,
```



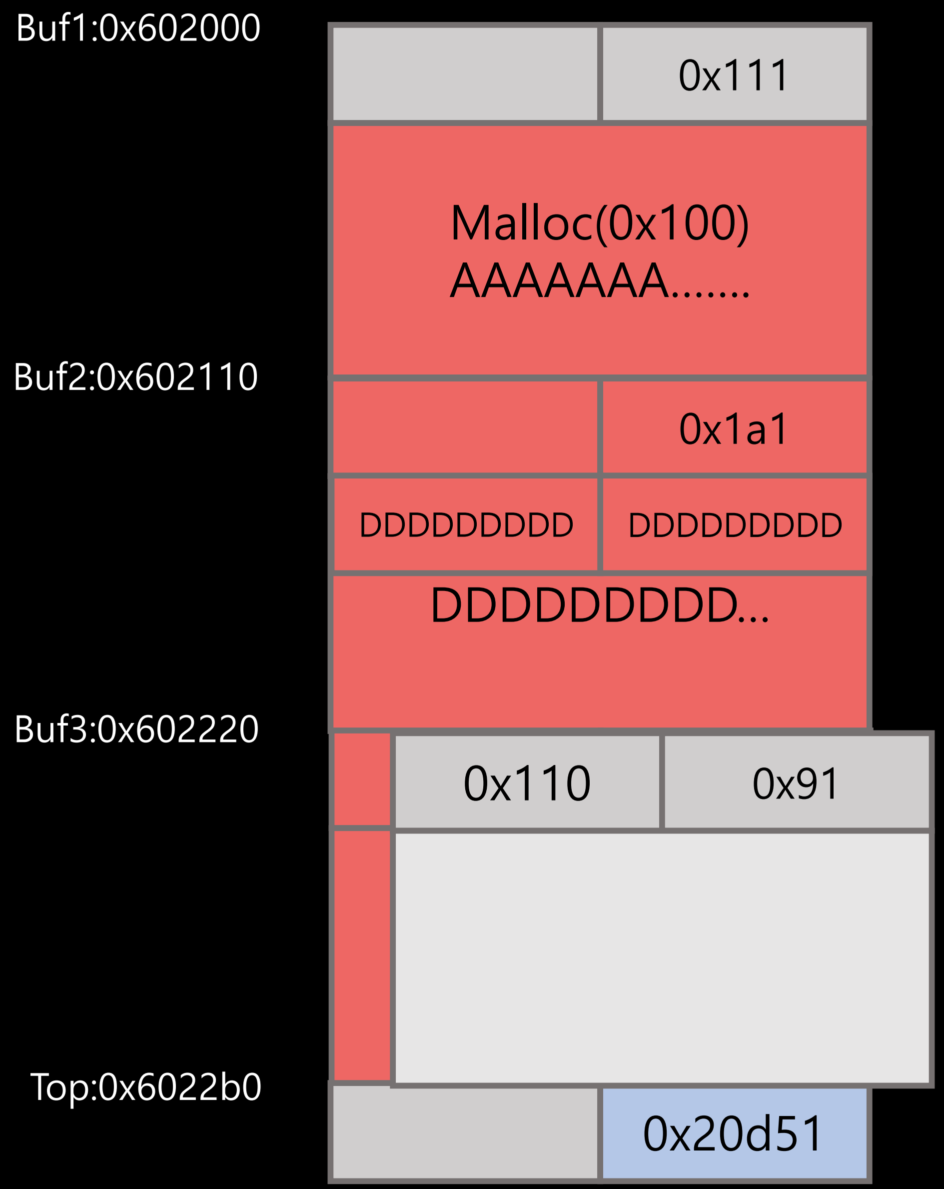
Overlapping chunks

```
gdb-peda$ x/54gx 0x00602110
0x602110:       0x0000000000000000      0x00000000000001a1
0x602120:       0x4444444444444444      0x4444444444444444
0x602130:       0x4444444444444444      0x4444444444444444
0x602140:       0x4444444444444444      0x4444444444444444
0x602150:       0x4444444444444444      0x4444444444444444
0x602160:       0x4444444444444444      0x4444444444444444
0x602170:       0x4444444444444444      0x4444444444444444
0x602180:       0x4444444444444444      0x4444444444444444
0x602190:       0x4444444444444444      0x4444444444444444
0x6021a0:       0x4444444444444444      0x4444444444444444
0x6021b0:       0x4444444444444444      0x4444444444444444
0x6021c0:       0x4444444444444444      0x4444444444444444
0x6021d0:       0x4444444444444444      0x4444444444444444
0x6021e0:       0x4444444444444444      0x4444444444444444
0x6021f0:       0x4444444444444444      0x4444444444444444
0x602200:       0x4444444444444444      0x4444444444444444
0x602210:       0x4444444444444444      0x4444444444444444
0x602220:       0x4444444444444444      0x4444444444444444
0x602230:       0x4444444444444444      0x4444444444444444
0x602240:       0x4444444444444444      0x4444444444444444
0x602250:       0x4444444444444444      0x4444444444444444
0x602260:       0x4444444444444444      0x4444444444444444
0x602270:       0x4444444444444444      0x4444444444444444
0x602280:       0x4444444444444444      0x4444444444444444
0x602290:       0x4444444444444444      0x4444444444444444
0x6022a0:       0x4343434343434300      0x4343434343434343
0x6022b0:       0x0000000000000000      0x0000000000020d51
```



Overlapping chunks

```
gdb-peda$
0x602110:       0x0000000000000000      0x00000000000001a1
0x602120:       0x4444444444444444      0x4444444444444444
0x602130:       0x4444444444444444      0x4444444444444444
0x602140:       0x4444444444444444      0x4444444444444444
0x602150:       0x4444444444444444      0x4444444444444444
0x602160:       0x4444444444444444      0x4444444444444444
0x602170:       0x4444444444444444      0x4444444444444444
0x602180:       0x4444444444444444      0x4444444444444444
0x602190:       0x4444444444444444      0x4444444444444444
0x6021a0:       0x4444444444444444      0x4444444444444444
0x6021b0:       0x4444444444444444      0x4444444444444444
0x6021c0:       0x4444444444444444      0x4444444444444444
0x6021d0:       0x4444444444444444      0x4444444444444444
0x6021e0:       0x4444444444444444      0x4444444444444444
0x6021f0:       0x4444444444444444      0x4444444444444444
0x602200:       0x4444444444444444      0x4444444444444444
0x602210:       0x4444444444444444      0x4444444444444444
0x602220:       0x4444444444444444      0x4444444444444444
0x602230:       0x4444444444444444      0x4444444444444444
0x602240:       0x4444444444444444      0x4444444444444444
0x602250:       0x4444444444444444      0x4444444444444444
0x602260:       0x4444444444444444      0x4444444444444444
0x602270:       0x4444444444444444      0x4444444444444444
0x602280:       0x4444444444444444      0x4444444444444444
0x602290:       0x4444444444444444      0x4444444444444444
0x6022a0:       0x4343434343434300      0x4343434343434343
0x6022b0:       0x0000000000000000      0x0000000000020d51
gdb-peda$ c
buf3 : DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDD
```



Q&A