

ICS Homework 12

May 20, 2021

1 Dynamic Linking

Given the PLT table:

```
PLT[1] <free>
400450: jmpq *0x20055a(%rip)
400456: pushq $0x0
40045b: jmpq 0x400440 PLT[2] <printf>
400460: jmpq *0x200552(%rip)
400466: pushq $0x1
40046b: jmpq 0x400440 ...

PLT[5] <malloc>
400490: jmpq *0x20053a(%rip)
400496: pushq $0x4
40049b: jmpq 0x400440
```

1. Please fill in the GOT table before the execution of main

Address	Entry	Contents	Description
0x600998	GOT[0]	--	
0x6009a0	GOT[1]	--	
0x6009a8	GOT[2]	--	
	GOT[3]		
	GOT[4]		
	GOT[7]		

2. We have the following code:

```
1: #include <stdlib.h>
2: void main(){
3:     int a = 1;
4:     printf("%d\n", a);
5:     char *c;
6:     c = (char*)malloc(4);
7:     printf("%d\n", a);
8:     free(c);
9: }
```

The addresses of functions are given:

printf	0x00007ffff7a81cf0
malloc	0x00007ffff7aacfc0
free	0x00007ffff7aad600

Which entry of the GOT table will change and what will it be after the execution of:

line 4:

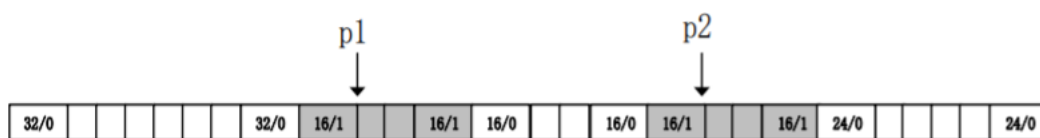
line 6:

line 7:

line 8:

2 Dynamic Memory Allocation

The figure simulates the initial status of memory at a certain time. Allocated blocks are shared, and free blocks are blank (each block represents 4 bytes). The allocator maintains double-word alignment. Given the execution sequence of memory allocation operations (malloc() or free()) from 1 to 4.



- 1: *P3 = malloc(4);*
- 2: *P4 = malloc(10);*
- 3: *free(P1);*
- 4: *P5 = malloc(16);*

Assume **first-fit** algorithm is used to find free blocks and coalesce immediately. Please draw the status of memory and mark with variables after the **2nd** and **4th** operation is executed.