Objectives for 10/30/2017 (Mon)

- Structure containing 2D array
  - How do allocate and deallocate using only one call (each) to malloc(…) and free(…))

264

Stack

party_shortcut_1.c

addr	type*	name*	value	part	fn
200	int	argc	1	args	main(...)
204	char**	argv	→ {"./foo"}		
212	void*			ret addr	
220	int	num_rows	4	locals	
	int	num_cols	4		
	Party*	party	400		
	int	i	4		

This was rewritten after class.

Heap

addr	value	
400	.num_people	50
404	.start_time	11
408	.name	600
416	.daze[0]	448
	.daze[1]	452
	.daze[2]	456
	.daze[3]	460
448	****	
452	****	
456	****	
460	****	
464		

Data segment

addr	type*	value
600	char*	"Birthday"

10-30-2017

- Type and name are not actually stored in memory or executable. Addresses shown are fictional.
- Assume `sizeof(int)==4` `sizeof(char)==1` `sizeof(void*)==8`
- To show struct types with fields, split the type and name fields. In value field, just write the value of the field. Example →

type	name	value
Point : int, : int	p . x . y	5 6

void*

void* is any memory address.

- malloc(...) returns void*.
- In C: you can assign a void* to a [ ]* variable.
- When you add to a void*, you are adding in bytes.

  Add to int*, increments by sizeof(int).