

CHAPTER 15

1. Run *relocation.py* with seeds 1, 2, and 3, and compute whether each virtual address generated by the process is in or out of bounds. If in bounds, compute the translation.

➔ Seed = 1

Virtual Address Trace (Limit: 290, base: 13884)			
Virtual Address	Decimal Rep.	Valid?	Decimal Translation?
VA 0	782	SEGMENTATION	-
VA 1	261	VALID	14145
VA 2	507	SEGMENTATION	-
VA 3	460	SEGMENTATION	-
VA 4	667	SEGMENTATION	-

➔ Seed = 2

Virtual Address Trace (Limit: 500, base: 15529)			
Virtual Address	Decimal Rep.	Valid?	Decimal Translation?
VA 0	57	VALID	15586
VA 1	86	VALID	15615
VA 2	855	SEGMENTATION	-
VA 3	753	SEGMENTATION	-
VA 4	685	SEGMENTATION	-

➔ Seed = 3

Virtual Address Trace (Limit: 316, base: 8916)			
Virtual Address	Decimal Rep.	Valid?	Decimal Translation?
VA 0	378	SEGMENTATION	-
VA 1	618	SEGMENTATION	-
VA 2	640	SEGMENTATION	-
VA 3	67	VALID	8983
VA 4	13	VALID	8929

2. Run with the following flags: *-s 0 -n 10*. What value do you have set *-l* (the bound register) to in order to ensure that all the generated virtual addresses are within bounds?
 - In order to ensure that all addresses are within bounds, we must set the '*-l*' flag to be equal to the biggest address value + 1; with 10 virtual addresses, the biggest address possible is 929. Therefore, the flag needs to be set to 930.

CHAPTER 16

1. First let's use a tiny address space to translate some addresses. Here's a simple set of parameters with a few different random seeds; can you translate the addresses?

- `segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s 0`

Virtual Address Trace [Segment 0 base: 0, limit 20; Segment 1 base: 512, limit 20]			
Virtual Address	Decimal Rep.	Valid?	Decimal Translation?
VA 0	108	Valid in SEG1	492
VA 1	97	Segmentation (SEG1)	-
VA 2	53	Segmentation (SEG0)	-
VA 3	33	Segmentation (SEG0)	-
VA 4	65	Segmentation (SEG1)	-

- `segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s 1`

Virtual Address Trace [Segment 0 base: 0, limit 20; Segment 1 base: 512, limit 20]			
Virtual Address	Decimal Rep.	Valid?	Decimal Translation?
VA 0	17	Valid in SEG0	17
VA 1	108	Valid in SEG1	492
VA 2	97	Segmentation (SEG1)	-
VA 3	32	Segmentation (SEG0)	-
VA 4	63	Segmentation (SEG0)	-

- `segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s 2`

Virtual Address Trace [Segment 0 base: 0, limit 20; Segment 1 base: 512, limit 20]			
Virtual Address	Decimal Rep.	Valid?	Decimal Translation?
VA 0	122	Valid in SEG1	506
VA 1	121	Valid in SEG1	505
VA 2	7	Valid in SEG0	7
VA 3	10	Valid in SEG0	10
VA 4	106	Segmentation (SEG1)	-

2. What is the highest legal virtual address in segment 0? What about the lowest legal virtual address in segment 1? What are the lowest and highest illegal addresses in this entire address space? Finally, how would you run `segmentation.py` with the `-A` flag to test if you are right?
 - The highest legal virtual address in segment 0 is 19; The lowest legal virtual address in segment 1 is 108
 - The highest illegal address in the whole address space is 107, and the lowest illegal address is 20
 - To test the lowest legal address, run: `./segmentation ... -A 20`
 - To test the highest legal address, run: `./segmentation ... -A 107`

3. Let's say we have a tiny 16-byte address space in a 128-byte physical memory. What base and bounds would you set up so as to get the simulator to generate the following translation results for the specified address stream: valid, valid, violation, ..., violation, valid, valid? Assume the following parameters: `segmentation.py -a 16 -p 128 -A 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 --b0 ? --l0 ? --b1 ? --l1 ?`
- b0: segment 0 base = 0
 - l0: segment 0 limit = 2
 - b1: segment 1 base = 128
 - l1: segment 1 limit = 2

CHAPTER 18

2. Now let's do some translations. Start with some small examples, and change the number of pages that are allocated to the address space with the `-u` flag. For example:
- `-P 1k -a 16k -p 32k -v -u 0`
 - `-P 1k -a 16k -p 32k -v -u 25`
 - `-P 1k -a 16k -p 32k -v -u 50`
 - `-P 1k -a 16k -p 32k -v -u 75`
 - `-P 1k -a 16k -p 32k -v -u 100`
 - What happens as you increase the percentage of pages that are allocated in each address space?
 - When the percentage of pages allocated in each address space increases, more memory access is possible, but at the cost of decreasing free space.