دانشکده‌ی مهندسی کامپیوتر

سیستم‌های عامل
تمرین‌های سری چهارم کارگاه

علی حیدری، محمدجواد میرشکاری

۱۵ خرداد ۹۸

فهرست مطالب

۱

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
segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s 1
segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s 2
```

پاسخ.

$$seed = 0 \quad \Rightarrow \quad VA_0 \quad \text{is valid}$$

```
ali@DESKTOP:~$ python2.7 segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s
    0
```

```
ARG seed 0
ARG address space size 128
ARG phys mem size 512

Segment register information:

  Segment 0 base  (grows positive) : 0x00000000 (decimal 0)
  Segment 0 limit                  : 20

  Segment 1 base  (grows negative) : 0x00000200 (decimal 512)
  Segment 1 limit                  : 20

Virtual Address Trace
  VA  0: 0x0000006c (decimal:  108) --> VALID in SEG1: 0x000001ec (decimal:  492)
  VA  1: 0x00000061 (decimal:   97) --> SEGMENTATION VIOLATION (SEG1)
  VA  2: 0x00000035 (decimal:   53) --> SEGMENTATION VIOLATION (SEG0)
  VA  3: 0x00000021 (decimal:   33) --> SEGMENTATION VIOLATION (SEG0)
  VA  4: 0x00000041 (decimal:   65) --> SEGMENTATION VIOLATION (SEG1)
```

$$seed = 1 \quad \Rightarrow \quad VA_0, VA_1 \quad \text{are valid}$$

```
ali@DESKTOP:~$ python2.7 segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s
    1 -c
ARG seed 1
ARG address space size 128
ARG phys mem size 512

Segment register information:

  Segment 0 base  (grows positive) : 0x00000000 (decimal 0)
  Segment 0 limit                  : 20

  Segment 1 base  (grows negative) : 0x00000200 (decimal 512)
  Segment 1 limit                  : 20

Virtual Address Trace
  VA  0: 0x00000011 (decimal:   17) --> VALID in SEG0: 0x00000011 (decimal:   17)
  VA  1: 0x0000006c (decimal:  108) --> VALID in SEG1: 0x000001ec (decimal:  492)
  VA  2: 0x00000061 (decimal:   97) --> SEGMENTATION VIOLATION (SEG1)
  VA  3: 0x00000020 (decimal:   32) --> SEGMENTATION VIOLATION (SEG0)
  VA  4: 0x0000003f (decimal:   63) --> SEGMENTATION VIOLATION (SEG0)
```

$$seed = 2 \quad \Rightarrow \quad VA_0, VA_1, VA_2, VA_3 \quad \text{are valid}$$

```
ali@DESKTOP:~$ python2.7 segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s
    2 -c
ARG seed 2
ARG address space size 128
ARG phys mem size 512

Segment register information:

  Segment 0 base  (grows positive) : 0x00000000 (decimal 0)
  Segment 0 limit                  : 20

  Segment 1 base  (grows negative) : 0x00000200 (decimal 512)
  Segment 1 limit                  : 20

Virtual Address Trace
```

```
  VA  0: 0x0000007a (decimal:  122) --> VALID in SEG1: 0x000001fa (decimal:  506)
  VA  1: 0x00000079 (decimal:  121) --> VALID in SEG1: 0x000001f9 (decimal:  505)
  VA  2: 0x00000007 (decimal:    7) --> VALID in SEG0: 0x00000007 (decimal:    7)
  VA  3: 0x0000000a (decimal:   10) --> VALID in SEG0: 0x0000000a (decimal:   10)
  VA  4: 0x0000006a (decimal:  106) --> SEGMENTATION VIOLATION (SEG1)
```

VA باید بین limit و ۰ یا Address Space و Address Space - limit باشد.

۲

Now, let's see if we understand this tiny address space we've constructed (using the parameters from the question above). 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?

پاسخ. VA باید بین limit و ۰ یا Address Space و Address Space - limit باشد. بنابراین بازه‌ی (۰،۲۰) و (۱۰۸،۱۲۸) مجاز است. کم‌ترین مقدار غیرقابل قبول ۲۰ و بیش‌ترین مقدار غیرقابل قبول ۱۰۷ است.

```
ali@DESKTOP:~$ python2.7 segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s
    0 -c
ARG seed 0
ARG address space size 128
ARG phys mem size 512

Segment register information:

  Segment 0 base  (grows positive) : 0x00000000 (decimal 0)
  Segment 0 limit                  : 20

  Segment 1 base  (grows negative) : 0x00000200 (decimal 512)
  Segment 1 limit                  : 20

Virtual Address Trace
  VA  0: 0x0000006c (decimal:  108) --> VALID in SEG1: 0x000001ec (decimal:  492)
  VA  1: 0x00000061 (decimal:   97) --> SEGMENTATION VIOLATION (SEG1)
  VA  2: 0x00000035 (decimal:   53) --> SEGMENTATION VIOLATION (SEG0)
  VA  3: 0x00000021 (decimal:   33) --> SEGMENTATION VIOLATION (SEG0)
  VA  4: 0x00000041 (decimal:   65) --> SEGMENTATION VIOLATION (SEG1)
```

```
ali@DESKTOP:~$ python2.7 segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s
    0 -c -A 20 -c
ARG seed 0
ARG address space size 128
ARG phys mem size 512

Segment register information:

  Segment 0 base  (grows positive) : 0x00000000 (decimal 0)
  Segment 0 limit                  : 20

  Segment 1 base  (grows negative) : 0x00000200 (decimal 512)
  Segment 1 limit                  : 20

Virtual Address Trace
  VA  0: 0x00000014 (decimal:   20) --> SEGMENTATION VIOLATION (SEG0)
```

```
ali@DESKTOP:~$ python2.7 segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s
    0 -c -A 107 -c
ARG seed 0
ARG address space size 128
ARG phys mem size 512

Segment register information:

  Segment 0 base  (grows positive) : 0x00000000 (decimal 0)
  Segment 0 limit                  : 20

  Segment 1 base  (grows negative) : 0x00000200 (decimal 512)
  Segment 1 limit                  : 20

Virtual Address Trace
  VA  0: 0x0000006b (decimal:  107) --> SEGMENTATION VIOLATION (SEG1)
```

۳

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 ?`

پاسخ. باید ۲ segment با ۲ limit تعریف کنیم.
Segment اول با Base 0 و Segment دوم با Base 16. باید توجه داشت که Segment دوم رشد منفی دارد.

```
ali@DESKTOP:~$ python2.7 segmentation.py -a 16 -p 128 -A
    0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 --b0 0 --l0 2 --b1 128 --l1 2 -c
ARG seed 0
ARG address space size 16
ARG phys mem size 128

Segment register information:

  Segment 0 base  (grows positive) : 0x00000000 (decimal 0)
  Segment 0 limit                  : 2

  Segment 1 base  (grows negative) : 0x00000080 (decimal 128)
  Segment 1 limit                  : 2

Virtual Address Trace
  VA  0: 0x00000000 (decimal:    0) --> VALID in SEG0: 0x00000000 (decimal:    0)
  VA  1: 0x00000001 (decimal:    1) --> VALID in SEG0: 0x00000001 (decimal:    1)
  VA  2: 0x00000002 (decimal:    2) --> SEGMENTATION VIOLATION (SEG0)
  VA  3: 0x00000003 (decimal:    3) --> SEGMENTATION VIOLATION (SEG0)
  VA  4: 0x00000004 (decimal:    4) --> SEGMENTATION VIOLATION (SEG0)
  VA  5: 0x00000005 (decimal:    5) --> SEGMENTATION VIOLATION (SEG0)
  VA  6: 0x00000006 (decimal:    6) --> SEGMENTATION VIOLATION (SEG0)
  VA  7: 0x00000007 (decimal:    7) --> SEGMENTATION VIOLATION (SEG0)
  VA  8: 0x00000008 (decimal:    8) --> SEGMENTATION VIOLATION (SEG1)
  VA  9: 0x00000009 (decimal:    9) --> SEGMENTATION VIOLATION (SEG1)
  VA 10: 0x0000000a (decimal:   10) --> SEGMENTATION VIOLATION (SEG1)
  VA 11: 0x0000000b (decimal:   11) --> SEGMENTATION VIOLATION (SEG1)
  VA 12: 0x0000000c (decimal:   12) --> SEGMENTATION VIOLATION (SEG1)
```

```
  VA 13: 0x0000000d (decimal:   13) --> SEGMENTATION VIOLATION (SEG1)
  VA 14: 0x0000000e (decimal:   14) --> VALID in SEG1: 0x0000007e (decimal:  126)
  VA 15: 0x0000000f (decimal:   15) --> VALID in SEG1: 0x0000007f (decimal:  127)
```

۴

Assume we want to generate a problem where roughly 90% of the randomly generated virtual addresses are valid (not segmentation violations). How should you configure the simulator to do so? Which parameters are important to getting this outcome?

پاسخ. برای این کار کافی است که Segmentهای ما روی هم 90% حافظه را بگیرند. مثلا اگر Address = 100 باشد کافی است که برای دو Segment خود یکی با Base 0 و یکی با Base 100 محدودیت‌های (limit) ۴۵ تایی در نظر بگیریم. توجه شود که Segment دوم رشد منفی دارد.

۵

Can you run the simulator such that no virtual addresses are valid? How?

پاسخ. limitها را صفر قرار می‌دهیم.

```
ali@DESKTOP:~$ python2.7 segmentation.py -b 0 -l 0 -B 0 -L 0 -c
ARG seed 0
ARG address space size 1k
ARG phys mem size 16k

Segment register information:

  Segment 0 base  (grows positive) : 0x00000000 (decimal 0)
  Segment 0 limit                  : 0

  Segment 1 base  (grows negative) : 0x00000000 (decimal 0)
  Segment 1 limit                  : 0

Virtual Address Trace
  VA  0: 0x00000360 (decimal:  864) --> SEGMENTATION VIOLATION (SEG1)
  VA  1: 0x00000308 (decimal:  776) --> SEGMENTATION VIOLATION (SEG1)
  VA  2: 0x000001ae (decimal:  430) --> SEGMENTATION VIOLATION (SEG0)
  VA  3: 0x00000109 (decimal:  265) --> SEGMENTATION VIOLATION (SEG0)
  VA  4: 0x0000020b (decimal:  523) --> SEGMENTATION VIOLATION (SEG1)
```