OS 2022fall 10.20 hw6

20307140008 施君豪

运行环境：Windows 10 (pycharm)

1. **第十六章**
2. **详见下列代码**

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

C:\Users\MichaelShi\Desktop\ostep-homework-dev-zhongyl\vm-segmentation>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) --> SEG1: 0x000001ec （PA）**

**VA 1: 0x00000061 (decimal: 97) --> segmentation violation**

**VA 2: 0x00000035 (decimal: 53) --> segmentation violation**

**VA 3: 0x00000021 (decimal: 33) --> segmentation violation**

**VA 4: 0x00000041 (decimal: 65) --> segmentation violation**

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

(base) C:\Users\Michael Shi\Desktop\ostep-homework-dev-zhongyl\vm-segmentation>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) --> SEG0: 0x00000011(PA)**

**VA 1: 0x0000006c (decimal: 108) --> SEG1: 0x000001ec (PA)**

**VA 2: 0x00000061 (decimal: 97) --> segmentation violation**

**VA 3: 0x00000020 (decimal: 32) --> segmentation violation**

**VA 4: 0x0000003f (decimal: 63) --> segmentation violation**

**（3）segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s 2**

(base) C:\Users\Michael Shi\Desktop\ostep-homework-dev-zhongyl\vm-segmentation>segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s 2

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) --> SEG1: 0x000001fa**

**VA 1: 0x00000079 (decimal: 121) --> SEG0: 0x000001f9**

**VA 2: 0x00000007 (decimal: 7) --> SEG0: 0x00000007**

**VA 3: 0x0000000a (decimal: 10) --> SEG0: 0x0000000a**

**VA 4: 0x0000006a (decimal: 106) --> segmentation violation**

1. **段0的最高合法虚拟地址：VA 0: 0x00000013 (decimal: 19)**

**段1的最低合法虚拟地址：VA 1: 0x0000006c (decimal: 108)**

**地址空间中最低的非法地址：0x00000014 (decimal 20)**

**地址空间中最高的非法地址：0x0000006b (decimal 107)**

**测试代码：segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s 1 -A 19,108,20,107**

(base) C:\Users\Michael Shi\Desktop\ostep-homework-dev-zhongyl\vm-segmentation>segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s 1 -A 19,108,20,107 -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: 0x00000013 (decimal: 19) --> VALID in SEG0: 0x00000013 (decimal: 19)

VA 1: 0x0000006c (decimal: 108) --> VALID in SEG1: 0x000001ec (decimal: 492)

VA 2: 0x00000014 (decimal: 20) --> SEGMENTATION VIOLATION (SEG0)

VA 3: 0x0000006b (decimal: 107) --> SEGMENTATION VIOLATION (SEG1)

1. **题意为除0，1，15，16有效外其余均为违反，故设置四个参数为：0，2，16，2**

**测试代码：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 16 --l1 2 -c**

**代码见下：**

(base) C:\Users\Michael Shi\Desktop\ostep-homework-dev-zhongyl\vm-segmentation>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 16 --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) : 0x00000010 (decimal 16)

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: 0x0000000e (decimal: 14)

VA 15: 0x0000000f (decimal: 15) --> VALID in SEG1: 0x0000000f (decimal: 15)

1. **由于各个位置出现概率均等，故有效空间占总空间90%即可。相关参数为-a，-b0，-l0，-b1，-l1。**

**测试代码举例：segmentation.py -a 10 -p 128 -A 0,1,2,3,4,5,6,7,8,9 --b0 0 --l0 5 --b1 10 --l1 4 -c**

(base) C:\Users\Michael Shi\Desktop\ostep-homework-dev-zhongyl\vm-segmentation>segmentation.py -a 10 -p 128 -A 0,1,2,3,4,5,6,7,8,9 --b0 0 --l0 5 --b1 10 --l1 4 -c

ARG seed 0

ARG address space size 10

ARG phys mem size 128

Segment register information:

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

Segment 0 limit : 5

Segment 1 base (grows negative) : 0x0000000a (decimal 10)

Segment 1 limit : 4

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) --> VALID in SEG0: 0x00000002 (decimal: 2)

VA 3: 0x00000003 (decimal: 3) --> VALID in SEG0: 0x00000003 (decimal: 3)

VA 4: 0x00000004 (decimal: 4) --> VALID in SEG0: 0x00000004 (decimal: 4)

VA 5: 0x00000005 (decimal: 5) --> SEGMENTATION VIOLATION (SEG1)

VA 6: 0x00000006 (decimal: 6) --> VALID in SEG1: 0x00000006 (decimal: 6)

VA 7: 0x00000007 (decimal: 7) --> VALID in SEG1: 0x00000007 (decimal: 7)

VA 8: 0x00000008 (decimal: 8) --> VALID in SEG1: 0x00000008 (decimal: 8)

VA 9: 0x00000009 (decimal: 9) --> VALID in SEG1: 0x00000009 (decimal: 9)

1. **-l0 和-l1的参数均设置为0即可。**

**测试代码举例：segmentation.py -a 10 -p 128 -A 0,1,2,3,4,5,6,7,8,9 --b0 0 --l0 0 --b1 10 --l1 0 -c**

(base) C:\Users\Michael Shi\Desktop\ostep-homework-dev-zhongyl\vm-segmentation>segmentation.py -a 10 -p 128 -A 0,1,2,3,4,5,6,7,8,9 --b0 0 --l0 0 --b1 10 --l1 0 -c

ARG seed 0

ARG address space size 10

ARG phys mem size 128

Segment register information:

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

Segment 0 limit : 0

Segment 1 base (grows negative) : 0x0000000a (decimal 10)

Segment 1 limit : 0

Virtual Address Trace

VA 0: 0x00000000 (decimal: 0) --> SEGMENTATION VIOLATION (SEG0)

VA 1: 0x00000001 (decimal: 1) --> SEGMENTATION VIOLATION (SEG0)

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 (SEG1)

VA 6: 0x00000006 (decimal: 6) --> SEGMENTATION VIOLATION (SEG1)

VA 7: 0x00000007 (decimal: 7) --> SEGMENTATION VIOLATION (SEG1)

VA 8: 0x00000008 (decimal: 8) --> SEGMENTATION VIOLATION (SEG1)

VA 9: 0x00000009 (decimal: 9) --> SEGMENTATION VIOLATION (SEG1)