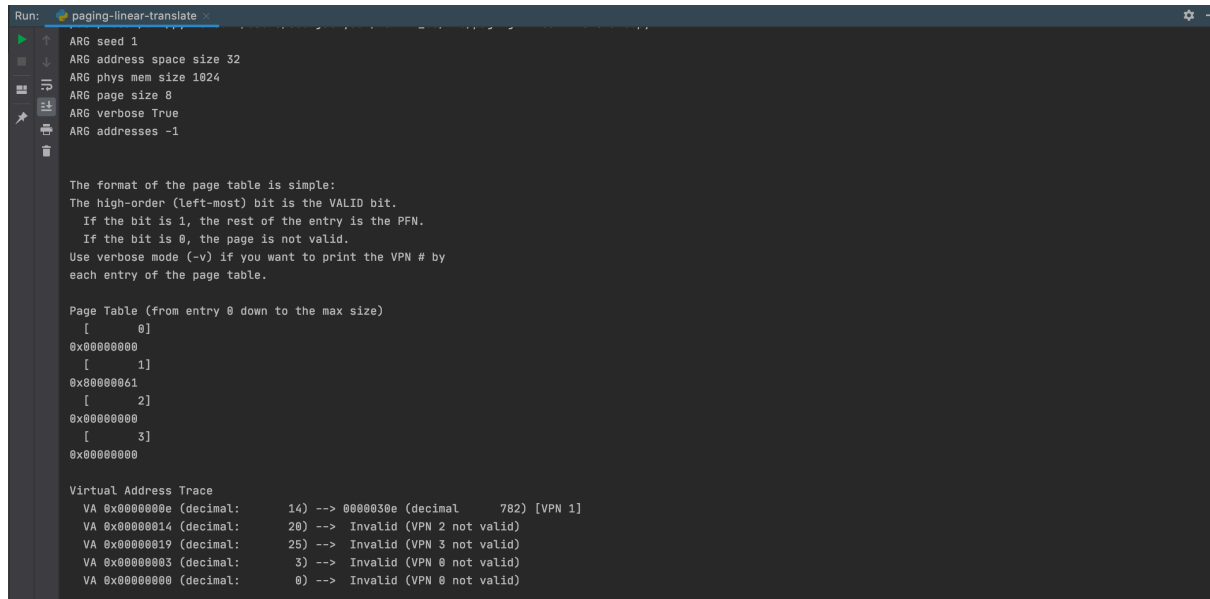


운영체제 Homework #3

201700949 설재혁

1.



```
Run: paging-linear-translate
ARG seed 1
ARG address space size 32
ARG phys mem size 1024
ARG page size 8
ARG verbose True
ARG addresses -1

The format of the page table is simple:
The high-order (left-most) bit is the VALID bit.
If the bit is 1, the rest of the entry is the PFN.
If the bit is 0, the page is not valid.
Use verbose mode (-v) if you want to print the VPN # by
each entry of the page table.

Page Table (from entry 0 down to the max size)
[ 0]
0x00000000
[ 1]
0x00000061
[ 2]
0x00000000
[ 3]
0x00000000

Virtual Address Trace
VA 0x0000000e (decimal: 14) --> 0000030e (decimal 782) [VPN 1]
VA 0x00000014 (decimal: 20) --> Invalid (VPN 2 not valid)
VA 0x00000019 (decimal: 25) --> Invalid (VPN 3 not valid)
VA 0x00000003 (decimal: 3) --> Invalid (VPN 0 not valid)
VA 0x00000000 (decimal: 0) --> Invalid (VPN 0 not valid)
```

첫 번째 옵션 결과

먼저 프로세스가 생성한 가상 주소의 변환을 위해 VPN과 Offset으로 분할을 해줘야 한다. 위 이미지에서 주소공간의 크기가 32이기 때문에 5bit가 필요하다. 페이지의 크기는 32byte 주소 공간 내에서 8byte이기 때문에 4개의 페이지를 선택할 수 있어야 하고 따라서 주소의 최상위 2bit가 VPN이 되고 나머지 3bit가 Offset이 된다. 그리고 페이지 테이블을 보면 VPN 1은 PFN 0x61에 매핑되어 있고 나머지는 전부 Invalid 상태인 것을 확인할 수 있다.

첫 번째 VA의 경우 십진수 14를 이진수로 나타내면 01110이 된다. 이 가상주소의 최상위 2bit는 VPN으로 나타내고 나머지 3개의 bit는 Offset을 나타내게 된다. 따라서 첫 번째 VA는 VPN 1의 6번째 바이트를 가리키고 있는 것이다.

두 번째 VA의 경우 20을 이진수로 바꾸면 10100이고 이 가상 주소는 VPN 2의 4번째 바이트가 된다. 하지만 VPN 2는 유효하지 않은 페이지이므로 이 주소는 유효하지 않은 주소가 된다.

세 번째 VA의 경우 25를 이진수로 바꾸면 11001이고 이 가상 주소는 VPN 3의 1번째 바이트가 된다. 마찬가지로 VPN 3은 유효하지 않은 페이지이므로 이 주소 또한 유효하지 않은 주소가 된다.

네 번째 VA의 경우 3을 이진수로 바꾸면 00011이고 이 가상 주소는 VPN 0의 3번째 바이트가 된다. 마찬가지로 VPN 0도 유효하지 않은 페이지이므로 이 주소 또한 유효하지 않은 주소가 된다.

마지막 VA의 경우 0을 이진수로 바꾸면 00000이고 이 가상 주소는 VPN 0의 0번째 바이트가 된다. 역시 VPN 0은 유효하지 않은 페이지이므로 이 주소 또한 유효하지 않은 주소가 된다.

앞서 확인한 바에 따라 Page Table을 보면 VPN 1은 PFN 0x61에 매핑이 되어 있는 것을 확인할 수 있다. 0x61을 이진수로 나타내면 1100001이고 VA의 Offset(110)을 뒤에다 붙이면 Physical Memory(실제 적재되는 메모리 주소)가 나오게 된다. 즉, 1100001110(Decimal:782, HexaDecimal:0000030e)가 된다.

이미지의 결과 또한 위와 동일한 것을 확인할 수 있다.

```

Run: paging-linear-translate
ARG seed 2
ARG address space size 32k
ARG phys mem size 1m
ARG page size 8k
ARG verbose True
ARG addresses -1

The format of the page table is simple:
The high-order (left-most) bit is the VALID bit.
If the bit is 1, the rest of the entry is the PFN.
If the bit is 0, the page is not valid.
Use verbose mode (-v) if you want to print the VPN # by
each entry of the page table.

Page Table (from entry 0 down to the max size)
[
  0]
0x00000079
[
  1]
0x00000000
[
  2]
0x00000000
[
  3]
0x0000005e

Virtual Address Trace
VA 0x000055b9 (decimal: 21945) --> Invalid (VPN 2 not valid)
VA 0x00002771 (decimal: 10097) --> Invalid (VPN 1 not valid)
VA 0x00004d8f (decimal: 19885) --> Invalid (VPN 2 not valid)
VA 0x00004dab (decimal: 19883) --> Invalid (VPN 2 not valid)
VA 0x00004a64 (decimal: 19044) --> Invalid (VPN 2 not valid)
  
```

두 번째 옵션 결과

위 이미지는 주소 공간의 크기가 32k이기 때문에 2^{15} 15bit가 필요한 것을 확인할 수 있다. 그리고 페이지의 크기가 8k이기에 4개의 페이지를 선택할 수 있어야 하고 따라서 15bit 주소의 최상위 2비트가 VPN 나머지 13bit가 Offset을 나타낸다.

Page Table을 보면 VPN 0이 0x79, VPN 3이 0x5e에 매핑이 되고 나머지 VPN 1, VPN 2는 Invalid 상태로 남겨진 것을 확인할 수 있다.

가상 주소를 물리적 주소로 변환하는 방법은 위와 동일하다.

첫 번째 VA = 0x000055b9(Hexadecimal) = 21945(Decimal) = 101010110111001(Binary)

➔ 최상위 2개의 bit가 VPN 2를 가리키므로 Invalid한 주소

두 번째 VA = 0x00002771 = 10097 = 010011101110001

➔ 최상위 2개의 bit가 VPN 1를 가리키므로 Invalid한 주소

세 번째 VA = 0x00004d8f = 19885 = 100110110101101

➔ 최상위 2개의 bit가 VPN 2를 가리키므로 Invalid한 주소

네 번째 VA = 0x00004dab = 19883 = 100110110101011

➔ 최상위 2개의 bit가 VPN 2를 가리키므로 Invalid한 주소

다섯 번째 VA = 0x00004a64 = 19044 = 100101001100100

➔ 최상위 2개의 bit가 VPN 2를 가리키므로 Invalid한 주소

5개의 VA 모두 미사용 Page를 가리키므로 유효하지 않은 주소다.

2.

```
Run: paging-linear-translate
ARG seed 3
ARG address space size 256m
ARG phys mem size 512m
ARG page size 1m
ARG verbose True
ARG addresses -1

The format of the page table is simple:
The high-order (left-most) bit is the VALID bit.
If the bit is 1, the rest of the entry is the PFN.
If the bit is 0, the page is not valid.
Use verbose mode (-v) if you want to print the VPN # by
each entry of the page table.

Page Table (from entry 0 down to the max size)
[
  0]
0x00000000
[
  1]
0x000000bd
[
  2]
0x00000140
[
  3]
0x00000000
[
  4]
0x00000000
[
  5]
0x00000084
[
  6]
0x00000000
[
  7]
0x000000f0

0x000000f5
[
  246]
0x000000ef
[
  247]
0x000001a4
[
  248]
0x000000f6
[
  249]
0x00000000
[
  250]
0x000001eb
[
  251]
0x00000000
[
  252]
0x00000000
[
  253]
0x00000000
[
  254]
0x00000159
[
  255]
0x00000000

Virtual Address Trace
VA 0x0308b24d (decimal: 58901581) --> 1f68b24d (decimal 526955885) [VPN 48]
VA 0x042351e6 (decimal: 69423590) --> Invalid (VPN 66 not valid)
VA 0x02feb67b (decimal: 58247291) --> 0a9eb67b (decimal 178173563) [VPN 47]
VA 0x0b46977d (decimal: 189175677) --> Invalid (VPN 180 not valid)
VA 0x0dbcccb4 (decimal: 230477492) --> 1f2cceb4 (decimal 523030196) [VPN 219]

Process finished with exit code 0
```

세 번째 옵션 결과

첫 번째 옵션의 경우 주소 공간의 크기가 32byte이기 때문에 너무나 작은 크기여서 Unrealistic하다고 생각한다.

세 번째 옵션의 경우 주소 공간의 크기가 256M인데 이는 너무 큰 크기이기 때문에 Unrealistic하다고 생각한다.

두 번째 옵션의 경우는 주소 공간의 크기가 32k인데 이는 조금은 작은 크기이긴 하지만 Unrealistic 할 정도는 아니라고 생각한다.

즉, 1, 3 번째 옵션을 실행했을 때가 Unrealistic한 파라미터의 조합이라고 생각한다.