CS 304 – Operating Systems
180010008
Assignment-3

LINEAR PAGE TABLE

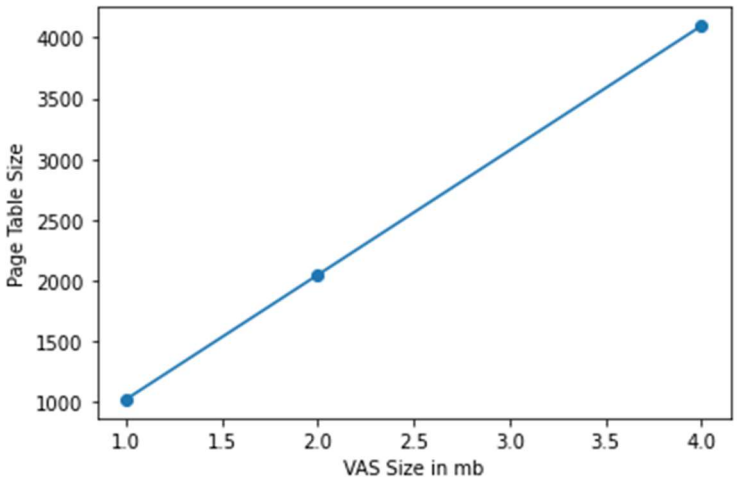
1. We know that,
   Page table size = (VAS Size / Page Size) * Size of 1 PTE
   Thus, Page Table Size should be ∝ Virtual Address Space Size.
   And Page Table Size should be 1/∝ Page Size.

   Now, lets run the **paging-linear-translate.py** and see the output for various cases.

   Case 1→ When VAS size is increased, what happens to the page table size.
   According to my analysis, Page table size is ∝ VAS size.

   Output for code:

S.No.	Flags given to program	Page Table Size
1.	-P 1k -a 1m -p 512m -v -n 0	1024
2.	-P 1k -a 2m -p 512m -v -n 0	2048
3.	-P 1k -a 4m -p 512m -v -n 0	4096

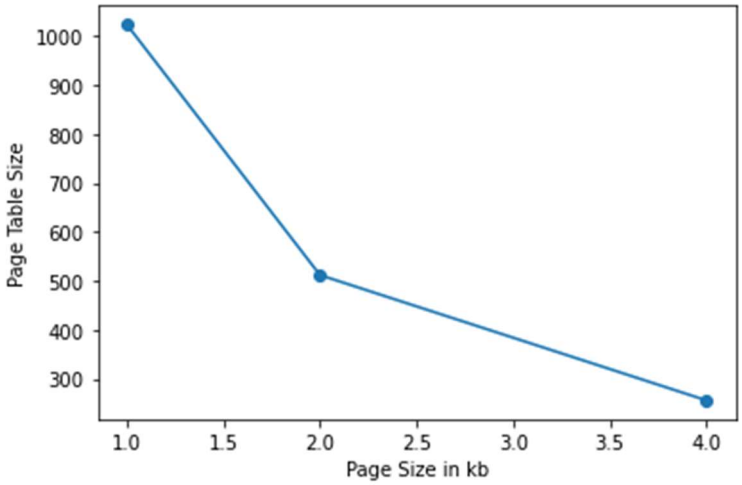


We can see that Page table size is increasing as VAS size increases.

Case 2→ When Page size is increased, what happens to the page table size.
According to my analysis, Page table size is 1/∝ Page size.

Output for code:

S.No.	Flags given to program	Page Table Size
1.	-P 1k -a 1m -p 512m -v -n 0	1024
2.	-P 2k -a 1m -p 512m -v -n 0	512
3.	-P 4k -a 1m -p 512m -v -n 0	256



We can see that Page table size is decreasing as Page size increases. Thus, the output result is consistent with the formula given above.

We shouldn't use bigger pages in general because they can cause internal fragmentation in the pages. We saw in the class slides that if we increase the page size, more space is wasted internally that leads to internal fragmentation. Applications end up allocating pages but only using little bits and pieces of each, and memory quickly fills up with these overly-large pages.

2. According to the README file,
   -u flag → percent of address space that is used.
   -u can take values from 0 → 100.
   0 means that no address space allocated to the process is used.
   100 means that full VAS allocated to the process is used.

a) When '-u == 0' → 0% address space used.

```
Virtual Address Trace
  VA 0x00003a39 (decimal:    14905) -->  Invalid (VPN 14 not valid)
  VA 0x00003ee5 (decimal:    16101) -->  Invalid (VPN 15 not valid)
  VA 0x000033da (decimal:    13274) -->  Invalid (VPN 12 not valid)
  VA 0x000039bd (decimal:    14781) -->  Invalid (VPN 14 not valid)
  VA 0x000013d9 (decimal:     5081) -->  Invalid (VPN 4 not valid)
```

b) When '-u == 25' → 25% address space used.

```
Virtual Address Trace
  VA 0x00003986 (decimal:    14726) -->  Invalid (VPN 14 not valid)
  VA 0x00002bc6 (decimal:    11206) --> 00004fc6 (decimal    20422) [VPN 10]
  VA 0x00001e37 (decimal:     7735) -->  Invalid (VPN 7 not valid)
  VA 0x00000671 (decimal:     1649) -->  Invalid (VPN 1 not valid)
  VA 0x00001bc9 (decimal:     7113) -->  Invalid (VPN 6 not valid)
```

c) When '-u == 50' → 50% address space used.

```
Virtual Address Trace
  VA 0x00003385 (decimal:    13189) --> 00003f85 (decimal    16261) [VPN 12]
  VA 0x0000231d (decimal:     8989) -->  Invalid (VPN 8 not valid)
  VA 0x000000e6 (decimal:      230) --> 000060e6 (decimal    24806) [VPN 0]
  VA 0x00002e0f (decimal:    11791) -->  Invalid (VPN 11 not valid)
  VA 0x00001986 (decimal:     6534) --> 00007586 (decimal    30086) [VPN 6]
```

d) When '-u == 75' → 75% address space used.

```
Virtual Address Trace
  VA 0x00002e0f (decimal:    11791) --> 00004e0f (decimal    19983) [VPN 11]
  VA 0x00001986 (decimal:     6534) --> 00007d86 (decimal    32134) [VPN 6]
  VA 0x000034ca (decimal:    13514) --> 00006cca (decimal    27850) [VPN 13]
  VA 0x00002ac3 (decimal:    10947) --> 00000ec3 (decimal     3779) [VPN 10]
  VA 0x00000012 (decimal:       18) --> 00006012 (decimal    24594) [VPN 0]
```

e) When '-u == 100' → 100% address space used.

```
Virtual Address Trace
  VA 0x00002e0f (decimal:    11791) --> 00004e0f (decimal    19983) [VPN 11]
  VA 0x00001986 (decimal:     6534) --> 00007d86 (decimal    32134) [VPN 6]
  VA 0x000034ca (decimal:    13514) --> 00006cca (decimal    27850) [VPN 13]
  VA 0x00002ac3 (decimal:    10947) --> 00000ec3 (decimal     3779) [VPN 10]
  VA 0x00000012 (decimal:       18) --> 00006012 (decimal    24594) [VPN 0]
```

0% of used address space leads to all entries invalid in VA → PA translation. Thus 0 is illegal to have.
As percentage of used address space is increased in terms of %, more and more VA → PA translations become valid in table.
Thus, we see as percentage of Address space used increases, more valid entries appear in the table and as it reaches 100%, all entries are valid.

3. From Q1, we know that,

   Page table size = (VAS Size / Page Size) * Size of 1 PTE
   Thus, **Page Size = (VAS Size / Page table size) * Size of 1 PTE**

S.No.	Command	Page Size	Physical Memory
1.	-P 8 -a 32 -p 1024 -v -s 1	4	1024
2.	-P 8k -a 32k -p 1m -v -s 2	4	1m
3.	-P 1m -a 256m -p 512m -v -s 3	256	512m

In this way, the first and third are unrealistic. The size of the third page table is too large, and the number of the first page table is not very large, but the page itself is too small, and 1024 corresponding to the actual physical address is too small to be appropriate.
Thus, first and third are unrealistic according to me.

4. Following are the errors/limitations of the given program →

- Here, I gave physical memory size less than address space size, so this gave an error.

```
sher@DELL-G3:/mnt/d/VI Sem CSE/CS 304 - Operating Systems/Assignment 3$ python2.7 paging-linear-translate.py -a 65k -v -c
ARG seed 0
ARG address space size 65k
ARG phys mem size 64k
ARG page size 4k
ARG verbose True
ARG addresses -1

Error: physical memory size must be GREATER than address space size (for this simulation)
```

- Here, I gave the address space size to be zero.

```
sher@DELL-G3:/mnt/d/VI Sem CSE/CS 304 - Operating Systems/Assignment 3$ python2.7 paging-linear-translate.py -a 0 -v -c
ARG seed 0
ARG address space size 0
ARG phys mem size 64k
ARG page size 4k
ARG verbose True
ARG addresses -1

Error: must specify a non-zero address-space size.
```

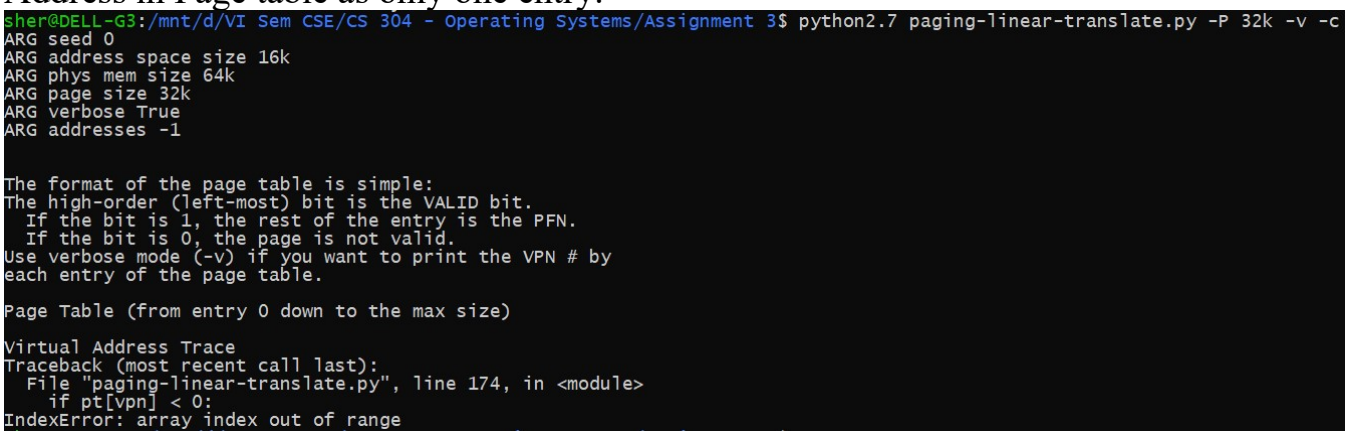
- Here, I gave the physical memory size to be zero.

```
sher@DELL-G3:/mnt/d/VI Sem CSE/CS 304 - Operating Systems/Assignment 3$ python2.7 paging-linear-translate.py -p 0 -v -c
ARG seed 0
ARG address space size 16k
ARG phys mem size 0
ARG page size 4k
ARG verbose True
ARG addresses -1

Error: must specify a non-zero physical memory size.
```

- Here, I gave the Page Size to be zero.

```
sher@DELL-G3:/mnt/d/VI Sem CSE/CS 304 - Operating Systems/Assignment 3$ python2.7 paging-linear-translate.py -P 0 -v -c
ARG seed 0
ARG address space size 16k
ARG phys mem size 64k
ARG page size 0
ARG verbose True
ARG addresses -1

Traceback (most recent call last):
  File "paging-linear-translate.py", line 85, in <module>
    mustbemultipleof(asize, pagesize, 'address space must be a multiple of the pagesize')
  File "paging-linear-translate.py", line 14, in mustbemultipleof
    if (int(float(bignum)/float(num)) != (int(bignum) / int(num))):
ZeroDivisionError: float division by zero
```

- Here, I made the page size so large that it gave an error in indexing Virtual Address in Page table as only one entry.



MULTI-LEVEL PAGE TABLE

1. With a linear page table, we need a single register to locate the page table, assuming that hardware does the lookup upon a TLB miss. We need **2 registers** to locate a two-level page table and **3 registers** to locate a three-level table.
2. Here, we are running this program for 3 different seeds:
   - Seed = 1 → We can see that there are 5 Translations to Physical Addresses, 4 Fault (Page table entry not valid) and 1 Fault (Page Directory entry not valid). Total to be 10.
     Each VA → PA translation at most takes 3 memory lookups.

```
PDBR: 17  (decimal) [This means the page directory is held in this page]

Virtual Address 6c74:
  --> pde index:0x1b [decimal 27] pde contents:0xa0 (valid 1, pfn 0x20 [decimal 32])
    --> pte index:0x3 [decimal 3] pte contents:0xe1 (valid 1, pfn 0x61 [decimal 97])
      --> Translates to Physical Address 0xc34 --> Value: 06
Virtual Address 6b22:
  --> pde index:0x1a [decimal 26] pde contents:0xd2 (valid 1, pfn 0x52 [decimal 82])
    --> pte index:0x19 [decimal 25] pte contents:0xc7 (valid 1, pfn 0x47 [decimal 71])
      --> Translates to Physical Address 0x8e2 --> Value: 1a
Virtual Address 03df:
  --> pde index:0x0 [decimal 0] pde contents:0xda (valid 1, pfn 0x5a [decimal 90])
    --> pte index:0x1e [decimal 30] pte contents:0x85 (valid 1, pfn 0x05 [decimal 5])
      --> Translates to Physical Address 0x0bf --> Value: 0f
Virtual Address 69dc:
  --> pde index:0x1a [decimal 26] pde contents:0xd2 (valid 1, pfn 0x52 [decimal 82])
    --> pte index:0xe [decimal 14] pte contents:0x7f (valid 0, pfn 0x7f [decimal 127])
      --> Fault (page table entry not valid)
Virtual Address 317a:
  --> pde index:0xc [decimal 12] pde contents:0x98 (valid 1, pfn 0x18 [decimal 24])
    --> pte index:0xb [decimal 11] pte contents:0xb5 (valid 1, pfn 0x35 [decimal 53])
      --> Translates to Physical Address 0x6ba --> Value: 1e
Virtual Address 4546:
  --> pde index:0x11 [decimal 17] pde contents:0xa1 (valid 1, pfn 0x21 [decimal 33])
    --> pte index:0xa [decimal 10] pte contents:0x7f (valid 0, pfn 0x7f [decimal 127])
      --> Fault (page table entry not valid)
Virtual Address 2c03:
  --> pde index:0xb [decimal 11] pde contents:0xc4 (valid 1, pfn 0x44 [decimal 68])
    --> pte index:0x0 [decimal 0] pte contents:0xd7 (valid 1, pfn 0x57 [decimal 87])
      --> Translates to Physical Address 0xae3 --> Value: 16
Virtual Address 7fd7:
  --> pde index:0x1f [decimal 31] pde contents:0x92 (valid 1, pfn 0x12 [decimal 18])
    --> pte index:0x1e [decimal 30] pte contents:0x7f (valid 0, pfn 0x7f [decimal 127])
      --> Fault (page table entry not valid)
Virtual Address 390e:
  --> pde index:0xe [decimal 14] pde contents:0x7f (valid 0, pfn 0x7f [decimal 127])
      --> Fault (page directory entry not valid)
Virtual Address 748b:
  --> pde index:0x1d [decimal 29] pde contents:0x80 (valid 1, pfn 0x00 [decimal 0])
    --> pte index:0x4 [decimal 4] pte contents:0x7f (valid 0, pfn 0x7f [decimal 127])
      --> Fault (page table entry not valid)
```

- Seed = 2 → We can see that there are 5 Translations to Physical Addresses, 3 Fault (Page table entry not valid) and 2 Fault (Page Directory entry not valid). Total to be 10.
  Each VA → PA translation at most takes 3 memory lookups.

```
PDBR: 122  (decimal) [This means the page directory is held in this page]

Virtual Address 7570:
  --> pde index:0x1d [decimal 29] pde contents:0xb3 (valid 1, pfn 0x33 [decimal 51])
    --> pte index:0xb [decimal 11] pte contents:0x7f (valid 0, pfn 0x7f [decimal 127])
      --> Fault (page table entry not valid)
Virtual Address 7268:
  --> pde index:0x1c [decimal 28] pde contents:0xde (valid 1, pfn 0x5e [decimal 94])
    --> pte index:0x13 [decimal 19] pte contents:0xe5 (valid 1, pfn 0x65 [decimal 101])
      --> Translates to Physical Address 0xca8 --> Value: 16
Virtual Address 1f9f:
  --> pde index:0x7 [decimal 7] pde contents:0xaf (valid 1, pfn 0x2f [decimal 47])
    --> pte index:0x1c [decimal 28] pte contents:0x7f (valid 0, pfn 0x7f [decimal 127])
      --> Fault (page table entry not valid)
Virtual Address 0325:
  --> pde index:0x0 [decimal 0] pde contents:0x82 (valid 1, pfn 0x02 [decimal 2])
    --> pte index:0x19 [decimal 25] pte contents:0xdd (valid 1, pfn 0x5d [decimal 93])
      --> Translates to Physical Address 0xba5 --> Value: 0b
Virtual Address 64c4:
  --> pde index:0x19 [decimal 25] pde contents:0xb8 (valid 1, pfn 0x38 [decimal 56])
    --> pte index:0x6 [decimal 6] pte contents:0x7f (valid 0, pfn 0x7f [decimal 127])
      --> Fault (page table entry not valid)
Virtual Address 0cdf:
  --> pde index:0x3 [decimal 3] pde contents:0x9d (valid 1, pfn 0x1d [decimal 29])
    --> pte index:0x6 [decimal 6] pte contents:0x97 (valid 1, pfn 0x17 [decimal 23])
      --> Translates to Physical Address 0x2ff --> Value: 00
Virtual Address 2906:
  --> pde index:0xa [decimal 10] pde contents:0x7f (valid 0, pfn 0x7f [decimal 127])
      --> Fault (page directory entry not valid)
Virtual Address 7a36:
  --> pde index:0x1e [decimal 30] pde contents:0x8a (valid 1, pfn 0x0a [decimal 10])
    --> pte index:0x11 [decimal 17] pte contents:0xe6 (valid 1, pfn 0x66 [decimal 102])
      --> Translates to Physical Address 0xcd6 --> Value: 09
Virtual Address 21e1:
  --> pde index:0x8 [decimal 8] pde contents:0x7f (valid 0, pfn 0x7f [decimal 127])
      --> Fault (page directory entry not valid)
Virtual Address 5149:
  --> pde index:0x14 [decimal 20] pde contents:0xbb (valid 1, pfn 0x3b [decimal 59])
    --> pte index:0xa [decimal 10] pte contents:0x81 (valid 1, pfn 0x01 [decimal 1])
      --> Translates to Physical Address 0x029 --> Value: 1b
```

- Seed = 3 → We can see that there are 8 Translations to Physical Addresses, 1 Fault (Page table entry not valid) and 1 Fault (Page Directory entry not valid). Total to be 10.
  Each VA → PA translation at most takes 3 memory lookups.

```
PDBR: 30  (decimal) [This means the page directory is held in this page]

Virtual Address 45b0:
  --> pde index:0x11 [decimal 17] pde contents:0xcd (valid 1, pfn 0x4d [decimal 77])
    --> pte index:0xd [decimal 13] pte contents:0xaf (valid 1, pfn 0x2f [decimal 47])
      --> Translates to Physical Address 0x5f0 --> Value: 14
Virtual Address 7075:
  --> pde index:0x1c [decimal 28] pde contents:0x91 (valid 1, pfn 0x11 [decimal 17])
    --> pte index:0x3 [decimal 3] pte contents:0x8c (valid 1, pfn 0x0c [decimal 12])
      --> Translates to Physical Address 0x195 --> Value: 16
Virtual Address 135c:
  --> pde index:0x4 [decimal 4] pde contents:0xe2 (valid 1, pfn 0x62 [decimal 98])
    --> pte index:0x1a [decimal 26] pte contents:0xa5 (valid 1, pfn 0x25 [decimal 37])
      --> Translates to Physical Address 0x4bc --> Value: 11
Virtual Address 3727:
  --> pde index:0xd [decimal 13] pde contents:0xa6 (valid 1, pfn 0x26 [decimal 38])
    --> pte index:0x19 [decimal 25] pte contents:0xdc (valid 1, pfn 0x5c [decimal 92])
      --> Translates to Physical Address 0xb87 --> Value: 13
Virtual Address 48c4:
  --> pde index:0x12 [decimal 18] pde contents:0xa7 (valid 1, pfn 0x27 [decimal 39])
    --> pte index:0x6 [decimal 6] pte contents:0xde (valid 1, pfn 0x5e [decimal 94])
      --> Translates to Physical Address 0xbc4 --> Value: 0d
Virtual Address 22bc:
  --> pde index:0x8 [decimal 8] pde contents:0xb5 (valid 1, pfn 0x35 [decimal 53])
    --> pte index:0x15 [decimal 21] pte contents:0xac (valid 1, pfn 0x2c [decimal 44])
      --> Translates to Physical Address 0x59c --> Value: 1b
Virtual Address 15ee:
  --> pde index:0x5 [decimal 5] pde contents:0xc3 (valid 1, pfn 0x43 [decimal 67])
    --> pte index:0xf [decimal 15] pte contents:0xd4 (valid 1, pfn 0x54 [decimal 84])
      --> Translates to Physical Address 0xa8e --> Value: 1c
Virtual Address 7bb9:
  --> pde index:0x1e [decimal 30] pde contents:0x7f (valid 0, pfn 0x7f [decimal 127])
      --> Fault (page directory entry not valid)
Virtual Address 1ebe:
  --> pde index:0x7 [decimal 7] pde contents:0xe8 (valid 1, pfn 0x68 [decimal 104])
    --> pte index:0x15 [decimal 21] pte contents:0x7f (valid 0, pfn 0x7f [decimal 127])
      --> Fault (page table entry not valid)
Virtual Address 4a10:
  --> pde index:0x12 [decimal 18] pde contents:0xa7 (valid 1, pfn 0x27 [decimal 39])
    --> pte index:0x10 [decimal 16] pte contents:0xe6 (valid 1, pfn 0x66 [decimal 102])
      --> Translates to Physical Address 0xcd0 --> Value: 0c
```

3. The addresses in these exercises appear to be random, so the miss rate would be quite high and it doesn't fit temporal locality or spatial locality, so will cause slow accesses. In the real world I'd expect a very high hit rate due to temporal and spatial locality (Otherwise the TLB would be a useless idea).