Week 05

Deadline : Saturday, October 14th 2017

Preparation before lab

1. In this lab, you will learn how virtual memory works with some C program
2. Login to your badak account
3. Change your directory to "work" and create new directory named “work05” inside “work” directory

```
$ cd work
$ mkdir work05
```

4. Move all the file attached at scele along with this file to your work05 directory. Hint : use WinSCP or Tunnels. The files you need to move :

   (1) vm-to-memory.c
5. Change your directory to work05

```
$ cd work05
```

Virtual Memory Address to Physical Memory Address

1. You are given a program **vm-to-memory.c**, try to understand it with information given from number 5 until 10.
2. Try to give it a go. Compile the program with gcc.

```
$ gcc vm-to-memory.c -o vm-to-memory
```

3. Try to run the program with additional argument of ‘Help.’ This command will show you how to run the program.

```
$ ./vm-to-memory Help
```

4. Now try to convert and load some Random Address.

```
$ ./vm-to-memory Convert 12345678

$ ./vm-to-memory Load 12345678
```

5. You can see that the result are ‘00000000’ and ‘0E’ respectively. This is because the convert method are not fully implemented yet. Your job for this week task is to complete this method, with information as follows:

   Consider a multi-level management scheme with the following format for virtual address:

Virtual Page # (10 bits)	Virtual Page # (10 bits)	Offset (12 bits)

   Virtual addresses are translated into physical addresses of the following form:

Physical Page # (20 bits)	Offset (12 bits)

   Page Table Entries (PTE) are 32 bits in the following format, stored in big-endian form in memory (i.e the MSB is the first byte in memory)

Physical page # (20 bits)	OS Defined (3 bits)	0	Large Page	Dirty	Accessed	No cache	Write Through	User	Writable	Valid

   Assume the base table pointer for the current user level process is **0x00200000**.

Physical Memory (All Values are in Hexadecimal)

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
00000000	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
00000010	1E	1F	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D
...																
00001010	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
00001020	40	03	41	01	30	01	31	03	00	03	00	00	00	00	00	00

00001030	00	11	22	33	44	55	66	77	88	99	AA	BB	CC	DD	EE	FF
00001040	10	01	11	03	31	03	13	00	14	01	15	03	16	01	17	00
...																
00002030	10	01	11	00	12	03	67	03	11	03	00	00	00	00	00	00
00002040	02	20	03	30	04	40	05	50	01	60	03	70	08	80	09	90
00002050	10	00	31	01	10	03	31	01	12	03	30	00	10	00	10	01
...																
00004000	30	00	31	01	11	01	33	03	34	01	35	00	43	38	32	79
00004010	50	28	84	19	71	69	39	93	75	10	58	20	97	49	44	59
00004020	23	03	20	03	00	01	62	08	99	86	28	03	48	25	34	21
...																
00100000	00	00	10	65	00	00	20	67	00	00	30	00	00	00	40	07
00100010	00	00	50	03	00	00	00	00	00	00	00	00	00	00	00	00
...																
00103000	11	22	00	05	55	66	77	88	99	AA	BB	CC	DD	EE	FF	00
00103010	22	33	44	55	66	77	88	99	AA	BB	CC	DD	EE	FF	00	67
...																
001FE000	04	15	00	00	48	59	70	7B	8C	9D	AE	BF	D0	E1	F2	03
001FE010	10	15	00	67	10	15	10	67	10	15	20	67	10	15	30	67
...																
001FF000	00	00	00	00	00	00	00	65	00	00	10	67	00	00	00	00
001FF010	00	00	20	67	00	00	30	67	00	00	40	65	00	00	50	07
...																
001FFFF0	00	00	00	00	00	00	00	00	10	00	00	67	00	10	30	65
...																
00200000	00	10	00	07	00	10	10	07	00	10	20	07	00	10	30	07
00200010	00	10	40	07	00	10	50	07	00	10	60	07	00	10	70	07
00200020	00	10	00	07	00	00	00	00	00	00	00	00	00	00	00	00
...																
00200FF0	00	00	00	00	00	00	00	00	00	1F	E0	07	00	1F	F0	07

...																

6. The address that is converted will not surpass the value INT_MAX to signed int.
7. Pay attention to Flag Valid in PTE, if the Flag Valid is 0, the convert method is automatically returns ‘Address Invalid’.
8. You may use the method in the code, but you may also add other methods if you think it’s necessary.
9. Try to implement method ***convert**!

   Bonus points if the method can accept conversion bigger than INT_MAX like 0xFFFFF005.
10. Here’s some test cases that can help you out:
    a. ./vm-to-memory Convert 00001047

       Output:

       ```
       -------------------
       Your Query: Convert 00001047
       Converting Virtual Memory of 00001047 to Physical Memory
       VM Address 00001047 has a Physical Memory Address of 00002047
       -------------------
       ```
    b. ./vm-to-memory Load 00001047

       Output:

       ```
       -------------------
       Your Query: Load 00001047
       Loading Address 00001047
       VM Address 00001047 has a Physical Memory Address of 00002047
       Content inside Physical Address 00002047 is '50'
       -------------------
       ```
    c. ./vm-to-memory Load 00C005FF

       Output:

       ```
       -------------------
       Your Query: Load 00C005FF
       Loading Address 00C005FF
       VM Address 00C005FF has a Physical Memory Address of 112201FF
       ```

Content inside Physical Address 112201FF is 'cannot be Determined, lookup error!'

Privacy Matters, Encryption and Digital Signature using GnuPG

1. Hash and sign your works so the other know it truly your works

```
$ sha1sum * > SHA1SUM

$ sha1sum -c SHA1SUM

$ gpg --sign --armor --detach SHA1SUM
```

2. verify the works

```
$ gpg --verify SHA1SUM.asc
```

3. create a tar ball. Tar is a way to create an archive file. You can ask uncle G for more information

```
$ cd ..

$ tar cvfj work05.tbj work05/
```

4. encrypt your files (work05.tbj)

```
$  gpg  --output  work05.tbj.gpg  --encrypt  --recipient  OSTEAM

--recipient your@email.com work05.tbj
```

*Use the same email as your Email input on GnuPG key generator.

5. copy the file to your github account, under the file week05/

```
$ cp work05.tbj.gpg ~/os172/week05/work05.tbj.gpg
```

6. change your directory to **os172/week05/**
7. remove file named ”**dummy**”
8. check whether there is a file named ”**work05.tbj.gpg**” if you dont find it, do the copy once more.
9. push the change to GitHub server
10. done!

Review Your Work

Dont forget to check your files/folders. After this lab, your current os172 folder should looks like:

```
os172
    key
        mypublickey1.txt
    log
        log01.txt
        log02.txt
        log03.txt
        log04.txt
        log05.txt
    SandBox
        <some_random_name>
    week00
        report.txt
    week01
        lab01.txt
        report.txt
        whyStudyOS.txt
        what-time-script.sh
    week02
        work02.tbj.gpg
            *work02
                *00-toc.txt
                *01-public-osteam.txt
                *02-ls-al.txt
                *03-list-keys1.txt
                *04-list-keys2.txt
                *hello.c
                *hello
                *status.c
                *status
                *loop.c
                *loop
                *exercise.c
                *exercise
                *SHA1SUM
                *SHA1SUM.asc
    week03
        work03.tbj.gpg
            *work03
                *.profile
                *sudo-explanation.txt
                *what-is-boot.txt
                *SHA1SUM
```

```
                        *SHA1SUM.asc
        week04
            work04.tbj.gpg
                *work04
                *01-public-osteam.txt
                *lab04.txt
                *global-char.c
                *global-char
                *local-char.c
                *local-char
                *open-close.c
                *open-close
                *write.c
                *write
                *result1.txt
                *result2.txt
                *demo-file1.txt
                *demo-file2.txt
                *demo-file3.txt
                *demo-file5.txt
                *00-pointer-basic.c
                *00-step-1
                *00-step-2
                *00-step-3
                *00-step-4
                *SHA1SUM
```

```
                    *SHA1SUM.asc
week05
          Work05.tbj.gpg
                    *vm-to-memory.c
                    *vm-to-memory
week06
          dummy
week07
          dummy
week08
          dummy
week09
          dummy
week10
          dummy
xtra
          dummy
```

keep in mind for every files/folders with wrong name, you will get penalty point. *means file that should be inside the archived file.

Note: "" means file should be inside the archived file.*