Version Wed, March 14th, 2017 (rev2)

Your Precious (Computer) Memory

1. In this tutorial, we will learn about Memory
2. First, please login to your *badak* account
3. Create new directory named `work` right where you first login and directory named `work05` inside that `work` directory

```
$ mkdir work
$ cd work
$ mkdir work05
$ cd work05
$ pwd
/home/fasilkom/mahasiswa/b/budi/work/work05    # example
```

4. Now, you must have these 2 files for this week's task:

   a) `06-memory.c`

      this is a demo program for this tutorial, for. You can get it in here

   b) `Makefile`

      for "automatic" compile. You can get it in here

   Put these files in `work05` directory

5. Compile the `06-memory` program

```
$ make
gcc 06-memory.c -o 06-memory -Xlinker  \
   -Map=06-memory.map
```

   after this, there should be files named `06-memory` and `06-memory.map`. Ignore it for now, we will play with them later.

```
$ ls
06-memory  06-memory.c  06-memory.map  Makefile
```

6. Now for the first task, let's play with program named `top`

```
$ top
```

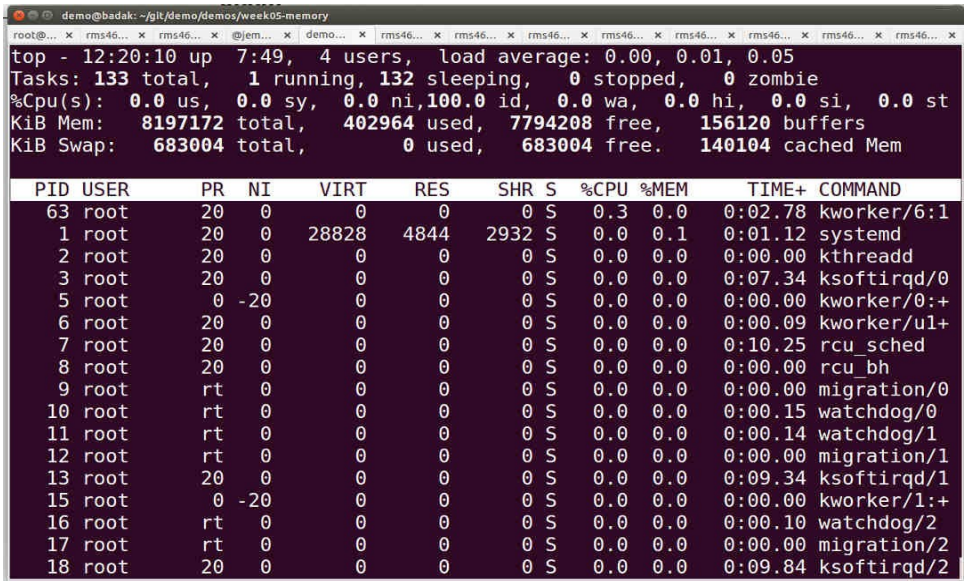

Illustration 1: top

**note: backup your `top` configuration if you already configure it by your preferences, because we will modify it for this task*

```
$ mv ~/.toprc ~/toprc.bak
```

In nut shell, `top` is a program that can listing the system processes (think it as *Task Manager* in *Windows*). And because you run it in *badak* server, it displaying the processes that running in *badak*.

In `top`, you can get interesting informations about the computer (in this case: badak server) like how many RAM installed, Memory usage, uptime server, user logged, etc.

The running processes displayed in table form. You can arrange the displayed column in `top`. Try to press "`f`" key to enter *Fields Management screen.*

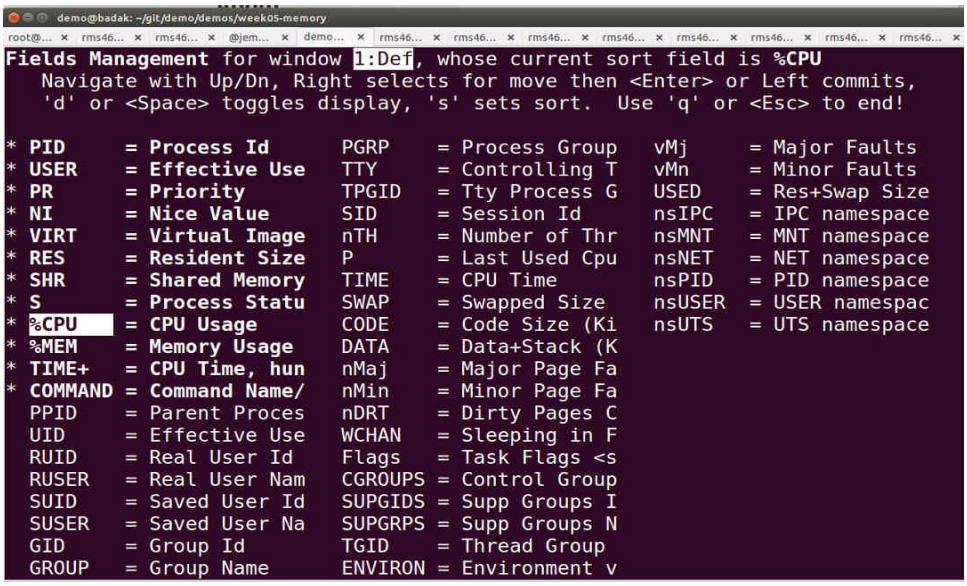

Illustration 2: Fields Management Screen

Try *to toggle the displayed column by pressing "`spacebar`" key.*

**hint: "*" symbol before name indicate it "on" or "displayed"*

To see the changes, exit from *Fields Management* screen by pressing "`q`" key

7. As the exercise, arrange the display so it only displaying the following columns:

   PID, PPID, %MEM, VIRT, RES, SHR, SWAP, CODE, DATA, USED.

   **note: the order is important. how to arrange the order I wonder? :)*

8. Save the configuration by exiting the fields management screen and press "`shift+w`" key (it will write the configuration to `~/.toprc` file)

9. Put the the configuration file to `work05` directory with name `toprc`

```
$ mv ~/.toprc toprc    # you can use cp if you want to keep it
```

First task is done, yay~

10. Now for the next task, we will play with the `06-memory` program that we compiled before

```
$ ./06-memory
```

    Quite familiar with the output? :)

11. For now, save the output to a file named `07-result.txt`

```
$ ./06-memory > 07-result.txt
```

12. As the second task, analyze the program (like what this program about, what did they do, etc). You can learn about it by reading the source code (`06-memory.c`)

    **hint: key points:* *`malloc()`*, *`system()`*

    Write your **own** analysis in file named `08-comments.txt`

Version Wed, March 14th, 2017 (rev2)

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, ensure you got OK for the all files

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

3. Create a tar ball and encrypt it

```
$ cd ..
$ tar cvfj work05.tbj work05
$ gpg --output work05.tbj.gpg --encrypt --recipient OSTEAM
work05.tbj
```

4. Copy the encrypted file to your repository, under the `week05` directory

```
$ cp work05.tbj.gpg ~/os171/week05
```

5. Change your working directory to `~/os171/week05/`

```
$ cd ~/os171/week05
```

6. Remove the "`dummy`" file

```
$ rm dummy
```

7. Check whether there is a file named "`work05.tbj.gpg`" or not
8. Do not forget to push the changes to GitHub server

Crafted with ♥ by ARA – Tested by AA

Review Your Work

Do not forget to check your files/directories. After this week task, your current os171 repository should looks like:

os171
- key
  - mypublickey1.txt
- log
  - <log_file>
  - …
- SandBox
  - <some_random_name>
  - …
- week00
  - report.txt
- week01
  - lab01.txt
  - report.txt
  - myExpectation.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
    - *01-public-osteam.txt
    - *.profile
    - *sudo-explanation.txt
    - *what-is-boot.txt
    - *SHA1SUM
    - *SHA1SUM.asc

week04

- work04.tbj.gpg
  - *work04
    - *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
  - *work05
    - *06-memory
    - *06-memory.c
    - *06-memory.map
    - *07-result.txt
    - *08-comments.txt
    - *Makefile
    - *SHA1SUM
    - *SHA1SUM.asc
    - *toprc

week06

- dummy

week07

- dummy

week08

- dummy

week09
    dummy
week10
    dummy
xtra
    dummy

keep in mind for every files/directories with wrong name/content, you will get penalty point.

note: "" means file that should be inside the archived file.*