

# COMP3000 - Exercise 1 (Individual)

## System Call Implementation

Winter 2018

In this exercise, you will implement a new MINIX POSIX-like system call with the signature:

```
void <your 1st name here>_time(int * sec, int * nsec);
```

for example `michel_time()`. It returns the system time using references to the two type `int` arguments `sec` and `nsec`, time in seconds and nanoseconds.

1. In the following, all file names are given relative to that path `/usr/src`.
2. Declare your system call in a new header file that you put in the directory `/include/`.
3. List the name of your file in the file `/usr/src/include/Makefile`.
4. Define your system call in a new `.c` file that you put in the directory `/minix/lib/libc/sys/`.
5. List the name of your file in the file `minix/lib/libc/sys/Makefile.inc`.
6. Define your system call number in the file `/minix/include/minix/callnr.h`.
7. Declare a function in file `/minix/servers/pm/proto.h`.
8. Register the function into the call vector in `/minix/servers/pm/table.c`, that has been reserved in file `/minix/include/minix/callnr.h`.
9. Define the function in a new `.c` file in the directory `/minix/servers/pm`.

Test your system call with the following program (must be updated with your 1st name).

```
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <michel_time.h>

int main(int argc, char** args) {
    int sec, nsec;
```

```

    /* get the current time */
    michel_time(&sec, &nsec);
    /* print local time */
    printf("System time is %d sec and %d nsec\n", sec, nsec);
    return 0;
}

```

Assuming source code is in file `tester.c`, sample output:

```

minix# cc -o tester tester.c
minix# ./tester
System time is 1508377450 sec and 4000000000 nsec

```

**Due date:** January 21. This exercise must be done in the C programming language under MINIX 3.4. Submit your work on cuLearn. Submit a single *tar.gz* file. Include a README.txt file containing a detailed report about your work (describe every change you made to the system code and where you made it), system call implementation and test program. You are responsible for the completeness of your submission. Source code and a make file must be included. Submissions that do not compile are not accepted. Here is an example start for the README.txt file:

Created file: `/usr/src/include/michel_time.h`

With:

```

/* Exercise 01 */
#ifndef _MICHEL_TIME_H_
#define _MICHEL_TIME_H_
void michel_time(int * sec, int * nsec);
#endif /* _MICHEL_TIME_H_ */

```

File name `michel_time.h` is listed in file `/usr/src/include/Makefile`.

...

Your submission must include a screenshot showing evidence that your code is working. In a single window, enter the following sequence of commands (see the attached example):

```
clear;uname -a;date; more tester.c;./tester.
```

Being meticulous is very important for that type of work. For debugging purposes, you may put `printf`-statements in your OS code. The output will be printed in the console and in file `/var/log/messages`.