

Operating Systems – Exercise 2

Processes, Multiprocessing & IPC

Submission & General Guidelines

- Submission deadline is **7/4/2022, 23:55** Moodle server time.
- This exercise must run properly on the docker environment.
- Submit your answers in the course website only as single **ex2-YOUR_ID.zip** (e.g. ex2-012345678.zip), containing **only**:
 - **ex2.c**
- Place your name and ID at the top of every source file, as well as in the PDF with the answers.
- No late submission will be accepted!
- Please give concise answers, but make sure to explain all answers.
- Write **clean code** (readable, documented, consistent, ...).

Part 1 (36 points)

In this question we will implement a C application that checks the response time of DNS queries to some internet domains.

The application should receive -

1. A path of a file containing a list of URLs (e.g. `http://google.com`)
2. The number of workers we want to use (e.g. 1)

After a successful run the application prints a numeric average of the successful name resolution queries, number of sites/domains reached, and the number of unknowns (for sites/domains it failed to test).

NOTICE: Set `CURLOPT_DNS_CACHE_TIMEOUT` to 0 to avoid OS DNS cache
(https://curl.se/libcurl/c/CURLOPT_DNS_CACHE_TIMEOUT.html)

You will implement it in both methods, PIPE and Memory mapping.

Add a flag “-f” in case you run it with pipe and without this flag the program run using memory mapping.

1. In addition to the manual pages from exercise 1, read the manual page for the following System Calls:
 - a. `FORK(2)` i.e. execute: `man 2 fork`
 - b. `PIPE(2)` also read `PIPE(7)`
 - c. `SIGNAL(7)`
 - d. `MMAP(2)`
 - e. `semaphore.h(0p)`

2. Install the libcurl development package on your VM:

```
sudo apt-get install libcurl3-dev
```

3. Create a new eclipse project for exercise 2, and change the following:
 - a. Project->Properties->C/C++ Build->Settings->Tool Settings->Cross GCC
Linker->Libraries->Add...->"curl" (do the same with "rt" and "pthread" instead of "curl")
4. Complete the application ex2.c (missing parts are marked with //TODO)

Examples (**numbers might vary between runs**)

```
$ ./ex2
usage:
    ./ex2 FILENAME NUMBER_OF_PROCESSES

$ ./ex2 top10.txt 1
0.0146 Average response time from 8 sites, 2 Unknown

$ ./ex2 top10.txt 2 -f
0.1002 Average response time from 9 sites, 1 Unknown

$ ./ex2 top10.txt 10 2
Illegal url detected, exiting now
```

Guidelines

- Use the manual.
- Make sure to always close the files you open.
 - Same goes for pipes.
- A parent must wait for all his/her child processes.
- Always check syscalls return value for errors. ALWAYS.
- You are not allowed to use any additional external libraries, if you are not sure if you are allowed to use something, ask in the forum.
 - You may use any function that was already used in the provided skeleton.
- You may assume that the function input is valid.
- Feel free to change the internals of given functions, but not their signatures.
- Notice that when compiling using gcc on the terminal, you should still use the "-lcurl" flag.
- Do **not** change the program output, i.e. the print statements.
- The check_url function returns a URL_ERROR if a URL address doesn't start with 'http'. If your program receives a URL_ERROR, all processes (parent and children) must exit immediately and the following message should be printed: "Illegal url detected, exiting now".
- **Hint:** a struct is just a bunch of bits, you can write() and read() structs.

Using Docker for Smoke Testing

- The VM is installed with Docker software (more on that later in the semester).
- When being graded, your solution will be tested in a container identical to the one that will be built using the supplied Dockerfile.
- There are two smoke tests (sanity checks) to this exercise. You can explore the 'EX2/check_submission/do_not_change' folder for more details
 - The first test checks your program against 10 international URLs using memory mapping
 - The second test checks your program against 8 Israeli URLs using pipe
- Usage:
 - You're given EX2 folder, containing:
 - A 'check_submission' folder, which contains a python script to execute (check_submission.py).
 - A skeleton file 'ex2.c'
 - Two text files: 'top10.txt', 'top128.txt'
 - Do not move ANY of the files inside check_submission, or you won't be able to use our tests.
 - You should edit 'ex2.c' with your solution.
 - **Each time** you want to test your final submission file (zip file):
 - Open a terminal and go the EX2/check_submission.
 - copy your submission file (ex2-YOUR_ID.zip) to your machine (anywhere you want)
 - Execute 'python check_submission.py <full path to your submission file>'If everything is fine, you will get:

```
student@student-VirtualBox:~/ex2/check_submission$ python check_submission.py /home/student/ex2/ex2-123456789.zip
A docker is being built. It might take a minute or two if this is the first time..
Done. Smoke tests are running now..
0.2002 Average response time from 10 sites, 0 Unknown
0.2461 Average response time from 8 sites, 0 Unknown
```

- Average response times may vary. Under some conditions one or two URLs may not respond; it is fine, but make sure that at least one of your runs is without unknown sites

Part 2 (24 points)

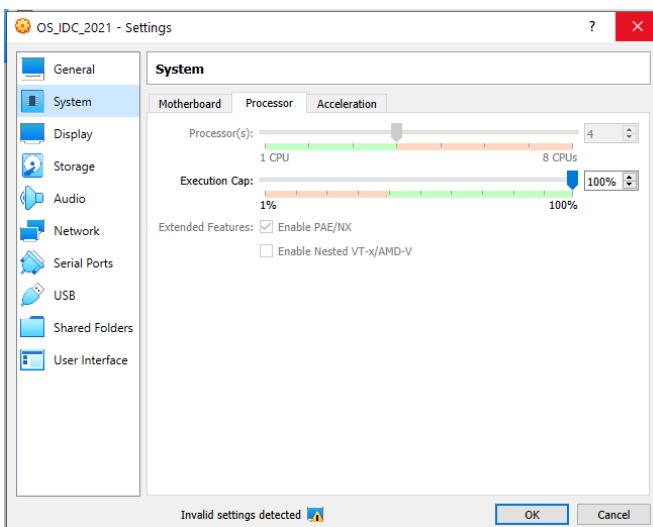
We will now examine the performance of our program from part 1.

1. Use the provided list of URLs: **top128.txt**
2. Run your program on this file, preceded with the **time(1)** command, using the following number of processes: 1, 2, 4, 8, 16, 32, 64, 128

```
$ time ./ex2 top128.txt 128
0.0949 Average response time from 116 sites, 12 Unknown

real 0m3.553s
user 0m1.076s
sys 0m0.120s
```

3. Draw a graph (using Google Sheets or Microsoft Excel) with a single series. The graph should show the **real** time each run takes (in milliseconds) [Y axis], as a function of the number of processes [X axis]. So it for both cases, memory mapping and pipe.
4. Explain the graph. Why are we seeing an improvement while only having a single CPU core in our VM? Why, when running a process for each URL the time is significantly larger than 2 seconds (the timeout for each URL check)?
5. Hypothetically, if we change the **check_url** function to only check if the structure of the URL is correct (i.e. `http://some.domain/path`), will the graph you draw change significantly? why?
 - **IMPORTANT:** Do not make this change. Submission with this change will result in 0 points for this question!
6. How will the results change if we use threads instead of processes (in c/Unix)? In your answer, please refer to overall time, memory use, etc. Explain briefly.
7. Change the VM configuration to 4 processors by clicking on “Settings” > “System” tab >



“Processor” tab, is there any change significant in the performance of your program? Please explain the results.

Part 3 (25 points)

What would be the output of the following pseudo C/UNIX code? Explain!

Notes:

Be careful not to explain what the code does! Do not tell a story about x, that it will be multiplied by 2 and then something else will happen.

Give a concise answer about what the output to the screen will be.

A. (10 points)

Consider two threads executing once the following C code, and accessing shared variables a, band c:

Initialization	Thread 1	Thread 2
Int a = 0; Int b = 0; Int c = 0;	a = 1 b = 5; if(a < 0 && b < 0) c = b - a;	a = -4; b = -1 if(a <= 0) c = a + b;

You can assume that each statement can be written using store and load operations (and these operations are not interrupted while running; they are atomic). What are the possible values for variable a,b, c after both threads complete?

B. (5 points)

Consider the following code:

```
int main()
{
    int x = 1;
    if (0 == fork()){
        int x = x + 1;
        fork();
    }
    else{
        int x = 4 * x;
        fork();
    }

    printf("%d ", x);
}
```

What will be the output if we run the given code? (the order does not matter)

C. (10 points)

When executing the following code, how many times will the line “forked” appear in the output?

```
int main()
{
    pid_t wpid;
    int status = 0;
    for ( int i=0; i<10; i++ ){
        if( i % 2 == 0){
            fork();
        }
    }
    printf("forked\n");
    // wait for all processes
    while ((wpid = wait(&status)) > 0);
    return 0;
}
```

Part 4 (15 points)

- A. Which of the following statement is FALSE?
- a. Threads within the same process share the same memory space
 - b. Process and its child process share the same memory space
 - c. CPU execute only one thread at a time
 - d. Processes can communicate through pipes
- B. Which of the following statement(s) regarding the difference between library calls and system calls is/are TRUE?
- a. A. Library calls are programming language dependent, while system calls are dependent only on the operating system.
 - b. B. System call may be invoked during a library call.
 - c. C. System calls are only invoked to handle system level I/O functionalities.
- a. (i) only.
 - c. (i) and (ii) only.
 - c. (ii) and (iii) only.
 - d. (i), (ii) and (iii).
 - e. None of the above.

For the statements below, state whether **it is true or false and explain concisely** (two lines at most):

- C. In kernel mode, you can read the memory of other process
- D. I've developed a multithreaded application that does a lot of HTTP requests, each HTTP request in its own thread. I've encountered performance issues and asked for help. My friend told me to use more processes because the CPU will trigger these requests faster. Was he right?