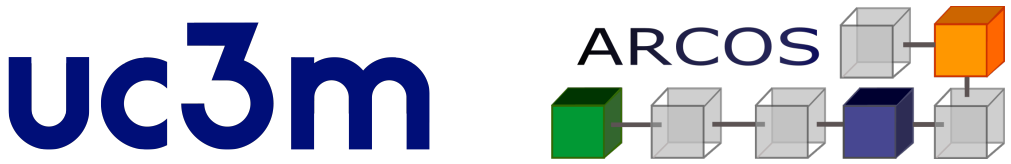


Computer Architecture and Technology Area

Universidad Carlos III de Madrid

OPERATING SYSTEMS
Lab 1. System Calls

Bachelor's Degree in Computer Science & Engineering
Bachelor's Degree in Applied Mathematics & Computing
Dual Bachelor in Computer Science & Engineering & Business Administration

Year 2021-2022

Contents

Contents

Lab Statement

- Implementation of three programs in C that are capable of manipulating files and directories.
- Each program will be implemented in just one file
  - **mycat.c** $\rightarrow$ It shows on the screen the content of a file
  - **myls.c** $\rightarrow$ It shows on the screen the entries of a directory
  - **mysize.c** $\rightarrow$ It shows the name and size of the entries of a directory that correspond to regular files

mycat

- `./mycat <path_to_input_file>`
- Result:
  - The program must show the **whole** content of a file on the screen.
  - The program must return -1 if no argument was introduced.
  - The program must return -1 if there was an error while opening the file (e.g. the file does not exist).
- Example:

```
1  $ ./mycat p1_tests/f1.txt
2  Name1    M    32    09834320    24500
3  Name2    F    35    32478973    27456
4  Name3    M    53    98435834    45000
```

myls

- `./myls <dir> //List dir`
- `./myls //List current dir`
- Result:
  - List of directory entries. **One entry per line**
  - The program must return -1 if there was an error while opening the file (e.g. the file does not exist).
- Example:

```
1  $ ./myls p1_tests/
2  dirC
3  f1.txt
4  dirA
5  f2.txt
6  .
7  ..
```

mysize

- `./mysize`
- Result:
  - List of regular files and their sizes.
  - The program will show **only** the data from regular files.
  - The program will show the data following this format:

    `<name><4_blank_spaces><size_in_bytes>`
  - The program should return -1 if there was an error opening the directory.
- Example:

```
1          $ cd p1_tests/
2          $ ../mysize
3          f1.txt     87
4          f2.txt     87
```

Initial Code

- A compressed folder including the files of the programs to be developed (*mycat.c*, *myls.c* and *mysize.c*) and a file to compile them (*Makefile*) is provided as the starting code.
- To compile: go to the folder and use the command `make`.

Tester

- Students are provided with the script in **python (Version 3)** **checker_os_p1.py**
- The tester must be executed in the Linux computers of the Virtual Aulas of the university.
- The command to execute the tester is the following:

```
python3 checker_os_p1.py <submitted_file.zip>
```

- Example:

```
$ python3 checker_os_p1.py
os_p1_100254896_100047014.zip
```

Contents

Assignment Submission & Rules

- Groups: 3 students maximum
- Delivery:
  - Source code in a compressed file
  - Lab report in PDF through TURNITIN
  - Only one member of the group may deliver
- Delivery date:

March 11th 2022 (until 23:55h)

Contents

System Calls

- For **mycat** you must use the calls related to file manipulation (open, read, write, and close).
- For **myls** you must use calls related to directory manipulation (getcwd, opendir, readdir and closedir).
- For **mysize** you must mix both types of calls.
- The documentation of those calls can be found in the manuals (see appendix).

Management of input arguments

- Extract the path of the files and directories that are passed as arguments to the program.
- **Implement error handling routines.**
- Example:

```
1   int main (int argc, char *argv[]){
2       int returnValue = 0;
3       if (argc >= 3){
4           printf(argv[0]);
5           printf(argv[1]);
6           printf(argv[2]);
7       }else{
8           printf("Not enough arguments\n");
9           returnValue = -1;
10      }
11      return returnValue;
12  }
```

Printing on the screen

- **mycat**
  - Write directly in the standard output the buffer that contains the data.

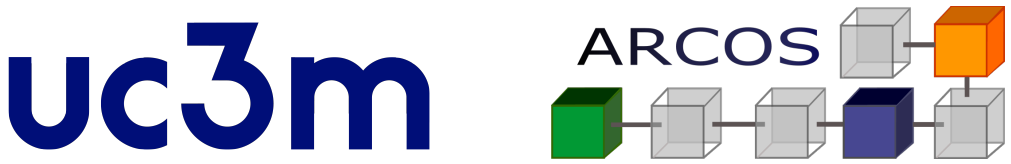
```
write(STDOUT_FILENO, buffer, SIZE_BUFFER);
```

- **myls**
  - Print with the call printf the field *d_name* of the structure *dirent*.

```
printf("%s\n", input->d_name);
```

- **mysize**
  - Print with the call *printf* the field *d_name* of the structure *dirent* and the obtained size, with four blank spaces between them.

```
printf("%s    %ld\n", input->d_name, tam);
```



Computer Architecture and Technology Area

Universidad Carlos III de Madrid

OPERATING SYSTEMS
Lab 1. System Calls

Bachelor's Degree in Computer Science & Engineering
Bachelor's Degree in Applied Mathematics & Computing
Dual Bachelor in Computer Science & Engineering & Business Administration

Year 2021-2022