

Operating Systems

3. Distance Learning 3 [18 Points]

Complete the tasks of this assignment on **our** (77.237.53.201) Ubuntu virtual machine.

Login to the following server:

IP: 77.237.53.201

Copy the following files from root (/) into your working (~/.distance3) directory on our virtual machine:

- /calc.sh
- /gencsv.sh

3.1. Linux Settings [3 Points]

Change the following settings on your virtual machine:

1. Create a new folder named “distance3” in your home directory. **[0,5 Points]**
2. Set the permissions of the directory “distance3” to owner = rwx, group = NONE and others = NONE. **[1 Point]**
3. Set the ownergroup of “solution.py” to *student*. **[0,5 Points]**
4. Set the permissions of the file “solution.py” to owner = rwx, group = x and others = NONE. Set the guid bit for the file. **[1 Point]**

3.2. Sort some file contents [15 Points]

For the following tasks you have to create a program in Python.

You can test your program by using the `check_solution.py` application

To test it, you have to type `“/check_solution.py ./solution.py”`

Create the files to be sorted by using the ***gencsv.sh*** script (type ***./gencsv.sh***).

This script generates 6 files “x-y.csv”, each containing 100 unsorted numbers x to y (1-100.csv to 501-600.csv).

Your task is to write a program “*solution.py*” (if using Python) that executes the following steps:

- read the content of each of the 6 generated files
- multiply each number in the files by 2, using the `calc.sh` script
- sort the numbers of each file by using the *slowsort*-algorithm implemented in our 2nd Assignment
- use the following pseudo-code to implement `calc.sh`, *slowsort* and *slowsort_start* in your code

```
procedure calc(value):
    returnvalue = exec("./calc.sh " + value)
    return returnvalue

procedure slowsort_start(A, i, j)
    for k=0 to len(A)
        arr[k] = calc(arr[k])
    slowsort(A, i, j)
```

- write out the multiplied and sorted numbers in files named “2x-2y.csv” (the unsorted file “1-100.csv” will be transformed to the sorted file “2-200.csv”, “101-200.csv” will result in “202-400.csv”, and so on).
- Change the permissions of each output file to “rw-----” (“600”)

Important:

- Your code has to use multi-threading: start *slowsort* or *slowsort_start* multi-threaded.
- You have a limited shared resource called “./calc.sh”. You can use this resource only twice in parallel. Use semaphores to get the right access to this resource.

The following rating scheme is used for the evaluation of your code:

3.2.1. Sorting and calculating [2 Points]

All lists must be sorted and calculated right.

This requirement is mandatory!

You will get 0 points for the whole task (3.2), if the lists are not sorted and calculated.

3.2.2. Coding Conventions [4 Points]

1. Your program has to be commented
2. Every function has to be commented
3. Every class, if any, has to be commented
4. The indentation style has to be consequent
5. Global variables have to be recognizable (prefix, caps lock, ...)
6. Variable names have to be in English

Check your Pythonscript by typing ***pylint solution.py*** . If you have a 10 out of 10 rating you get 3 Points.

3.2.3. Multithreading [4 Points]

Slowsort or ***slowsort_start*** have to be started multithreaded.

3.2.4. Shared Ressource [3 Points]

You have a limited shared resource called “***./calc.sh***”. You can use this resource only twice in parallel. Use semaphores to get the right access to this resource.

3.2.5. File permissions [2 Points]

The file permissions of your output files have to be set to “***rw-----***” (600).

3.3. Submission

To finalize your assignment, you have to submit the following documents:

1. Upload the **source code** of your program on moodle.
2. Show us your solution on the final exam.