Activity IX - Filesystem Implementation

Prepared by Krerk Piromsopa, Ph.D.

1. Filesystem in userspace (FUSE)

Fuse is an implementation of filesystem in userspace. When mounted, it bridges related system calls to functions in userspace. Thus, it is possible to create a (virtual) filesystem without modifying a kernel using libfuse.
To make your life easier, we will use python-fuse. To install, use **sudo apt install python3-fuse.**

a. Read-only InfoFS

Download **MyFS.py** from https://goo.gl/QJRfvu

In this filesystem, the related methods (getattr, readdir, open, and read) have been implemented. Create a folder mnt and mount the filesystem.
$mkdir mnt
$chmod +x MyFS.py
$./MyFS.py -o uid=1000 -o gid=1000 mnt

Note that you might want to replate uid and gid with your information. (You may use the '**id**' command to determine your uid and gid.)

Use the **ls (ls -li)** command or the **stat (stat and stat -f)** command to display the information of the mounting point and related file.

To umount the filesystem without the root permission, use: **$fusermount -u mnt**

Answer the following questions in myCourseVille

- What is the role of fuse? Please briefly explain fuse and its' functions.
- What is the name of the filesystem used by the mounting point?
- Is it possible to mkdir, to copy, to move, and to delete a file in this mounting point? Please provide your analysis.

Checkpoint #1:
Modify the MyFS.py to add two additional files (/instructors and /students). For /instructors, the content shall be the name of instructors in this class. For /students, the content shall be your names. Show this to your instructors for checking. Please also submit your code to CourseVille.

Here is the expected output.

```
krerk@OSBox:~/mnt $ cat instructors
0:CP ENG CU OS 2018S1 - Instructors
1:    Thongchai Rojkangsadan
2:    Veera Muangsin, Ph.D.
3:    Krerk Piromsopa, Ph.D.

krerk@OSBox:~/mnt$ cat students
0:CP ENG CU OS 2018S1 – Students, Group Name: [groupname]
1: 4123456721 [member 1]
2: 4123456721 [member 2]
```

b. WebService FS

In this exercise, we will use FUSE to implement a filesystem that will invoke a web service. Please create a filesystem that contains the following functions:

- Contains a file name participation.
- A read to the participation file will retrieve data from a webpage and show the contain as an output. (GET from "**https://mis.cp.eng.chula.ac.th/krerk/teaching/2022s2-os/status.php")**
- An append to the participation file will post a data to **"https://mis.cp.eng.chula.ac.th/krerk/teaching/2022s2-os/checkIn.php"**

To ease understanding, the following code snippets demonstrate the read and write from the participation file respectively.

```
def myRead:
      req=requests.get('https://mis.cp.eng.chula.ac.th/krerk/teaching/2022s2-os
/status.php');
      content = bytes(req.text)
      return content;
```

```
def myWrite(buf):
      raw=buf.split(':')
      checkInUrl='https://mis.cp.eng.chula.ac.th/krerk/teaching/2022s2-os/check
In.php'parms= { 'studentid' : raw[0], 'name' : raw[1], 'email' : raw[2] }
      rpost=requests.post(checkInUrl, data=parms);
      return len(buf)
```

Checkpoint #2:

Demonstrate the following actions.

- View particiation file (using the cat command) to show the current content
- Append your studentid, name, and email to participation file (use echo “studentid:name:email” >> particiation
- View particiation file again to show that your information has been added.

Here is the expected output.

```
krerk@OSBox:~/mnt$ cat participation
2018-11-01 15:55:08     38xxxxx Krerk Piromsopa,Ph.D.   Krerk.P@chula.ac.th
161.200.192.89
krerk@OSBox:~/mnt$ echo “41xxxx:Student Good:Student.G@student.chula.ac.th” >>
participation
krerk@OSBox:~/mnt$ cat participation
2018-11-01 15:55:08     38xxxxx Krerk Piromsopa,Ph.D.   Krerk.P@chula.ac.th
161.200.192.89
2018-11-01 15:55:30     41xxxxx Student Good
Student.G@student.chula.ac.th    161.200.192.89
```

Please also submit your code to CourseVille.