Oregon State University
Ecampus

CS361: Assignment 1: Microservices Warm-Up

Overview

To demonstrate you can implement the microservices architecture, write software comprised of **three separate programs**:

1. A program that generates pseudo-random numbers (**PRNG Service**)
2. A program that, given a non-negative integer *i*, returns the i^{th} image in a set (order doesn't matter) (**Image Service**)
   - If *i* is >= the number of images, modulo *i* by the size of the image set
3. A user interface (**UI**) that either has a button or can receive a user command. When the button is pushed or the command is entered...
   (a) UI calls the PRNG Service
   (b) UI calls the Image Service using the pseudo-random number from the PRNG Service
   (c) UI displays the image (or a path to it)

The programs can be written in **any language(s)**.

Use **any set of images** (e.g., downloaded from https://www.kaggle.com/). **Store images locally in a folder**; no API calls needed. No DB needed.

How long will this assignment take?

It could take you **anywhere from 1 hour to 5+ hours** to complete this assignment depending on your comfort and familiarity with the programming language you choose to use.

Requirements

- UI must either have a button (if UI is graphical) or be able to receive a user command (if UI is text-based)
- Each of the three programs must run in a **different process**
- Programs must **NOT call each other** directly (e.g., do not import one program into another)
- As the **communication pipe**, use text files as follows:
  1. UI calls PRNG Service by writing the word "run" to prng-service.txt
  2. PRNG Service reads prng-service.txt, erases it, and writes a pseudo-random number to it

3. UI reads prng-service.txt to get the pseudo-random number
4. UI writes the pseudo-random number to image-service.txt
5. Image Service reads image-service.txt, erases it, and writes an image path to it
6. UI reads image-service.txt then displays the image (or path) to the user

Instructions

PART 1: Plan

After reading through the requirements and instructions, how long do you think it will take to complete the assignment and how will you go about it? Answer the following questions to start planning.

Complete each item below by replacing the highlighted text (**Usability note**: double-click the text to select it).

1. Which **programming language** will you use to complete this assignment?

   Python

2. **How familiar and comfortable** are you with this programming language? For example, when was the last time you used it?

   I am very familiar with Python having learned it years ago and using it every now and then. I most recently used Python for coding in an algorithms course.

3. **Split the implementation into several tasks** (at least three), **estimate** how long each will take, and then **schedule** when you will complete each task.

Task description	**Estimate of how long it will take to complete**	**Time block during which you'll complete the task**
Refresh on how to read and write files.	0.5 hours	4/2 10:00AM to 4/2 10:30AM
Write prng and imgsrv files	0.5 hours	4/2 10:30 AM to 4/2 11:00 AM
Write UI.py	1 hours	4/2 11:00 AM to 4/2 12:00 PM

4. Based on the above and other factors you think are relevant, **how long do you think it'll take to complete the entire assignment**? **Explain** your answer. Track how long it actually takes to complete the assignment (you'll be asked about this in Part 3).

   I think it will take me 2 hours to complete the entire assignment. I will have to research methods for reading and writing to files then use that to write the prng and imgsrv files. I will also run some tests on the two files individually to make sure they are working. I think writing the UI python code will be a bit more extensive than the two components.

Now that you have a time estimate, I recommend you double it! That'll help give you a time buffer in case you run into issues.

PART 2: Create Video

- Create a **short video** (5 minutes or less) demonstrating you have satisfied the implementation requirements. It is recommended that you aim for a video length of 2 minutes or under, to keep it concise. The upper limit is still 5 minutes, but you don’t need to use the whole 5 minutes.

PART 3: Reflect

Now that you’ve completed the assignment...

How long did it **actually** take you to complete the assignment?

2 hours and 15 minutes.

This reflection is meant to help you understand how accurate your time estimates tend to be so that you can factor that information into future time estimates. You won’t be asked to track your time spent on future assignment but you might find it helpful to do so on your own

Submission

- Attach PDF or Word document. You MUST follow the instructions at **Modules > 'HOW TO: Attach a Document to "Text Entry" Field'.**
- Embed or link to **video**. You MUST follow the instructions at **Modules > “HOW TO: Create and Upload a Video”.**

Grading

You are responsible for satisfying all criteria listed in the Canvas rubric for this assignment. You will be able to revise this assignment if you miss points.

Questions?

Please ask via Ed so that others can benefit from the answers.