Assignment

First Student code, LastName
Second Student code, LastName
Third Student code, LastName
Fourth Student code, LastName

Operating Systems
Jefferson A.
Cali, 2024-II

Assignment

General Description

The following outlines the guidelines for a theoretical-practical activity for students in the Operating Systems course. The activity specifically assesses their mastery of some basic concepts, definitions related to CPU management, processes, scheduling, and the implementation of a simulator using C or C++.

Objectives During the development of the activities, students will achieve the following:

- Enumerate the milestones in the evolution of computer systems.
- Describe the objectives and functions of modern operating systems.
- Simulate the behavior of the CPU and process management.
- Apply the principles of scheduling.

Before You Begin

Read the following:

- *Operating System Concepts* (9th Edition) by Silberschatz and Galvin: Chapters 1, 2, and 3
- *Modern Operating Systems* by Tanenbaum: Chapters 1 and 2

Create a git repository and add the link:

Use this document as template [here]

Activity No. 1: Conceptualization [40%]

After completing the recommended readings in this document, answer the following questions in your own words, concisely and accurately.

1. Enumerate the milestones in the evolution of computer systems.
2. What are the four components of a computer system? Describe each one.
3. What is the difference between a monolithic kernel and a microkernel?
4. Define an Operating System from two different perspectives.
5. What is the purpose of system calls?
6. What is a multiprogrammed operating system?
7. What is a process?
8. What are the states of a process?
9. What information is stored in the Process Control Block (PCB) associated with a process?
10. What are the main activities of an operating system in relation to process management?

Activity No. 2: Remembering the C and C++ languages [50%]

1. Implements the programs and functions below:

Write a program in c to find the leap year.
Write a progran to calculate factorial of a number.
Write a program in c to calculate power using recursion.
Write a program in c to find even or odd numbers.
Write a program in c to print fabonnaci series.
Write a Function to check uppercase letter.
Write a program in c to function to check lowercase letter
Find the greater of the three numbers
Write a program in c to type casting implicit explicit.
Write program to display number 1 to 10 in octal, decimal and hexadecimal system.

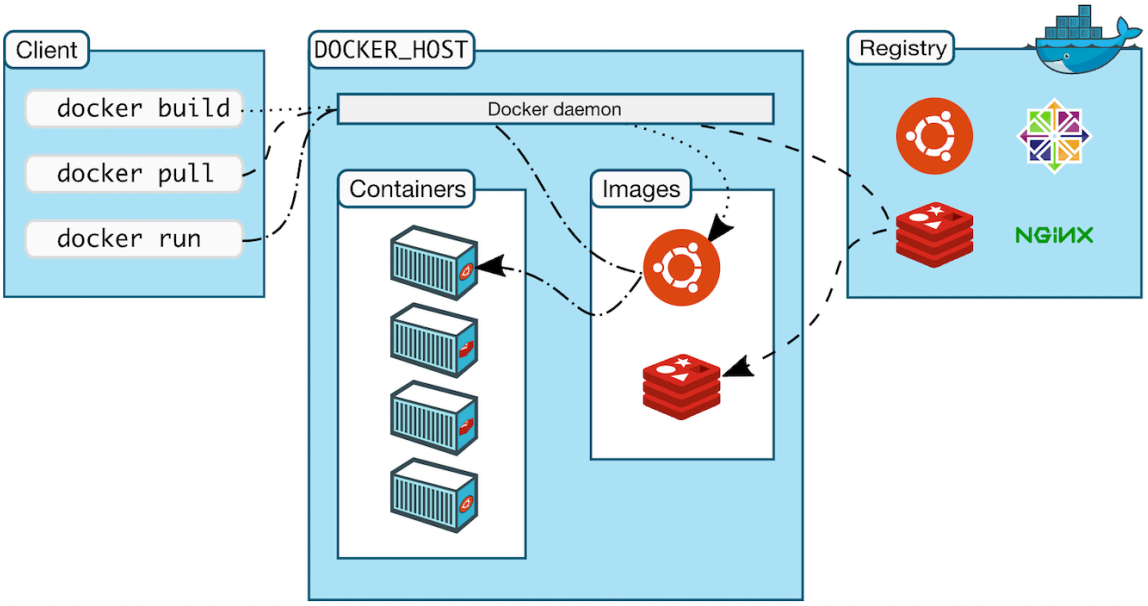
2. Write a C program that receives and processes grades for the Operating Systems (SO) course using structures.
3. Write a C++ program that performs the following tasks: First, the program should receive a numeric value and determine whether it is a prime number, displaying the result. Second, the program should accept a list of numbers and identify the prime numbers within that list. Lastly, the program should allow the user to input a numeric range and display all prime numbers within that specified range.
4. Create a program with C++ to model geometric shapes and perform area, perimeter, and color operations. Rectangle, square, triangle, etc. shapes must be considered. Tip. Use classes and inheritance.

Activity No. 3: Docker practical session [10%]

Docker is a command line-based software allowing users to manipulate images and create application containers.

As presented in the course, Docker consists of two elements:

- a client, to receive commands from the user
- a server, to execute commands and manage images and containers



Docker commands architecture

Typing this command will give the Client and Server versions available on your computer. ***Paste a screenshot of result***

docker *version*

Usage, options and a full list of available commands can be accessed through the command line in a terminal.Type the following command

docker *--help*

The general usage of a Docker command line is as follows:

docker [OPTIONS] COMMAND [arg...]

Questions

1. How many arguments are absolutely required by the command 'docker pull' ?
2. Do you remember what a registry is?

Download a predefined image available on the DockerHub

In a web browser, navigate to the DockerHub : https://hub.docker.com/
In the top search bar, type : japeto/pujgcc and paste a screenshot

Our course has images available for the development of practical sessions.

Questions

1. How many times was the *japeto* image downloaded ?

Execute the command inside a terminal.
Paste a screenshot of result

docker pull japeto/pujgcc:v0.12

You will get an error as this image has no default tag (“latest”). So we need to specify one in the command line.

Go to the “Tags” tab and copy the pull command of version latest
Paste a screenshot of result

docker pull japeto/pujgcc:v0.12

Question:

1. How many times do you see ‘Pull complete’ displayed ? Why ?

Now, to be sure that the image was correctly pulled, let’s see the list of all available downloaded images inside our workspace. ***Paste a screenshot of result***

docker image

Question:

1. What is the size of the japeto/pujgcc image ?

Perform a task using a pulled image

Among the Docker commands, we will now use the ‘run’ command.

Question:

1. What are the options and parameters of the ‘run’ command ?

docker run --help

As displayed in the terminal, the description of the command is 'Run a command in a new container'.

Question :

1. What is the difference between an image and a container ?

Now, to run the application, execute the following command:
Paste a screenshot of result

docker run japeto/pujgcc bash --help

Congratulations!
You just successfully downloaded and used your first Docker image!

Running *PUJGCC* without parameters was interesting as a demonstration of Docker's features. But if we want to really run *PUJGCC*, we also need to provide parameters and, most importantly, input files.

Find the paths to bind
To bind our current folder to the /data/ folder located inside a container, we first need the absolute path of the current folder, obtained through the unix pwd command.
Paste a screenshot of result

pwd

This path will be used in further commands through ${PWD}.

Instead of running ls command to /home/ files, we will now just list the content of the /data/ folder inside the container but bind with the host.
Paste a screenshot of result

docker run japeto/pujgcc:latest ls /data

If nothing appears, it is normal: the folder is empty and only serves as a "*branching point*".

We now have the paths of the two folders we want to bind together.

Bind a local folder into a container

To perform the folder mapping between the current folder and /data inside the image, the syntax is simple. ***Paste a screenshot of result***

docker run -v ${PWD}:/data/ japeto/pujgcc:latest ls /data/

Question:

1. Is the displayed list the same as what is in your current folder?

Finally, we can run C on a C or C++ file located in the Data folder. Change the name of the file to any of the provided files.
Paste a screenshot of result

docker run -v ${PWD}:/data/ japeto/pujgcc:latest gcc /data/helloworld.c

Restart and detach a container

Learn how to re-use a container where you installed something

Use the start command to restart the container created in the last exercise
Paste a screenshot of result

docker start -ti mycontainer /bin/bash

Go back to the container using the exec command instead of the run command.
Paste a screenshot of result

docker exec -ti mycontainer /bin/bash

Question :

1. *What happens now when you exit the container? Is it stopped?*

Check with and ***Paste a screenshot of result***

```
docker ps -l
```

In fact, the container keeps on running. This is because re-starting a container turns it into a "detached process" running in the background. Alternatively, we could have added the -d option to the first docker run command, creating directly a detached container.

Finally, you can stop the container.

```
docker stop mycontainer
```

This is the end of the practical session. We hope you enjoyed it. Don't hesitate to ask any questions and feel free to contact us any time after the session!

List of commands

Search the available versions of an image in the Docker registry:

```
docker search
```

Pulling an image:

```
docker pull
```

Starting a container on a given image running a single command:

```
docker run -ti
```

Starting a container on a given image running a single command (detached):

```
docker run -d
```

List all containers and their status

```
docker ps -l
```

List all pulled images

```
docker images
```

Removing one local container

```
docker rm
```

Removing one local image

```
docker rmi
```

Clean all containers

```
docker rm $(docker ps -aq)
```

Clean all images (after cleaning the containers)

```
docker rmi $(docker images -aq)
```

Observations

- Deliveries must be made in teams of 4. Using a public repository on github and a pdf report
- If you do not understand the instructions for any of the activities, do not hesitate to jefferson.amado.pena@correounivalle.edu.co