

Actividad Sistemas Operativos

Rui Yu Lei Wu

Julián Andrés Galvis

Actividad 1

1.

- **Primera Generación:** El hito principal es la válvula de vacío (bombilla de vacío). En esta generación no existían los sistemas operativos, sino que los operarios de máquinas realizaban manualmente las tareas.
- **Segunda Generación:** El hito principal es la aparición del transistor y las tarjetas perforadas. Los operarios siguen siendo lo más cercano a un sistema operativo, ya que manejan las tareas sin un sistema automatizado.
- **Tercera Generación:** El hito principal es la creación de circuitos integrados. Durante esta generación se crea el primer sistema operativo GNU.
- **Cuarta Generación:** El hito más importante es la aparición del ordenador personal (PC). Debido al alto costo de UNIX, surge el sistema operativo LINUX como una alternativa.

2.

- **Hardware:** Es la parte física del sistema, incluyendo procesadores, memoria, discos duros, etc.
- **Kernel:** Responsable de manejar las interacciones entre el hardware y el software.
- **Shell:** Es la interfaz entre el usuario y el kernel, permitiendo la comunicación entre ambos.
- **Aplicaciones:** Son los programas de los clientes.

3.

- **Kernel Monolítico:** Inicia todas las aplicaciones al encender la máquina, cargando todos los servicios necesarios desde el principio.
- **Microkernel:** Realiza una ejecución perezosa, donde las aplicaciones y servicios se inician solo cuando se necesitan.

4. Tanenbaum describe dos definiciones para los sistemas operativos, Top Down en la que el sistema operativo es una extensión de la máquina, y Bottom up en la que él es un administrador de los recursos de la máquina.

5. Las llamadas al sistema permiten al usuario interactuar directamente con el sistema operativo, indicándole qué tareas realizar.

6. Un sistema operativo multiprogramado permite que múltiples programas se ejecuten aparentemente a la vez, compartiendo recursos del sistema.

7. Un proceso tiene muchas definiciones validas, sin embargo, la más significativa es la siguiente: “Un proceso es un programa en ejecución”.

8.

- **Bloqueado:** El proceso no puede continuar hasta que se resuelva algún evento externo, como la finalización de una operación de I/O.
- **Listo (Ready):** El proceso está preparado para ejecutarse cuando se le asigne la CPU.
- **Run:** El proceso está siendo ejecutado activamente por la CPU.

9. El PCB almacena el ID del proceso, su estado actual, el contexto del proceso (incluyendo registros y el contador de programa), y las partes del proceso, como el código, el heap, y la pila (stack).

10. Las actividades principales incluyen la creación y eliminación de procesos, la suspensión y reanudación de procesos, la asignación de recursos necesarios para la ejecución de procesos, y la gestión de los estados de los procesos (bloqueado, listo, corriendo).

Actividad 5

Docker version

```
Client:
Version:      27.0.3
API version:  1.46
Go version:   go1.21.11
Git commit:   7d4bcd8
Built:        Sat Jun 29 00:03:32 2024
OS/Arch:      windows/amd64
Context:      desktop-linux

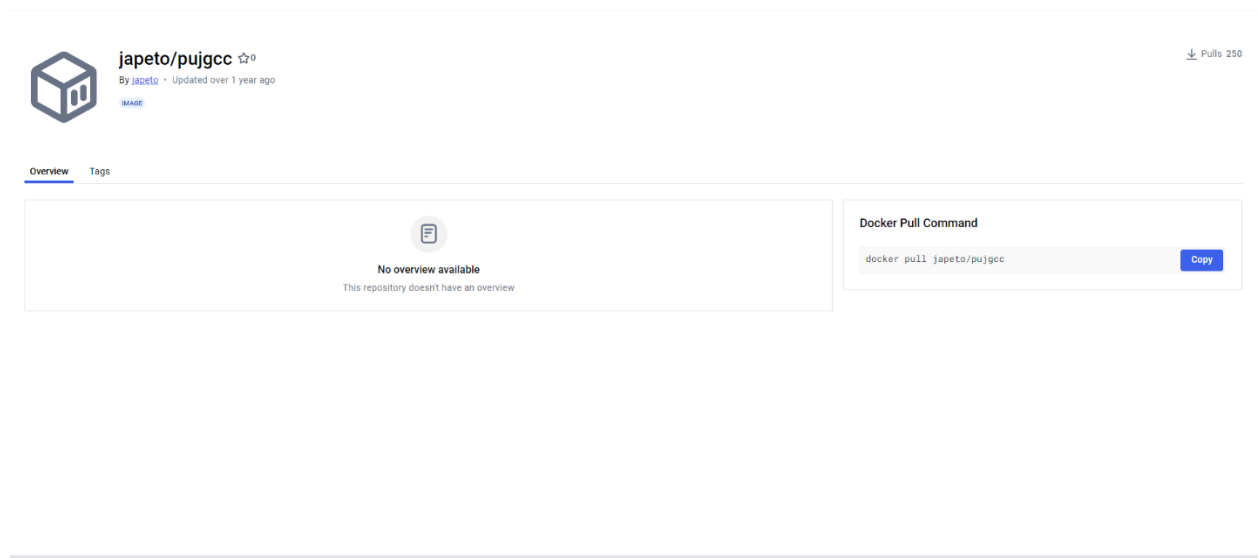
Server: Docker Desktop 4.32.0 (157355)
Engine:
Version:      27.0.3
API version:  1.46 (minimum version 1.24)
Go version:   go1.21.11
Git commit:   662f78c
Built:        Sat Jun 29 00:02:50 2024
OS/Arch:      linux/amd64
Experimental: false
containerd:
Version:      1.7.18
GitCommit:    ae71819c4f5e67bb4d5ae76a6b735f29cc25774e
runc:
Version:      1.7.18
GitCommit:    v1.1.13-0-g58aa920
docker-init:
Version:      0.19.0
GitCommit:    de40ad0
PS C:\Users\lab> |
```

1. How many arguments are absolutely required by the command ‘docker pull’?

El comando Docker pull requiere de al menos un argumento obligatoriamente, que sería la imagen a la que se hará el pull.

2. Do you remember what a registry is?

Un “registry” en Docker es un servicio que almacena imágenes de Docker, estos permiten a los usuarios descargar imágenes y subir imágenes para otros usuarios.



1. How many times was the japeto image downloaded ?

Se instala 11 veces.

```
PS C:\Users\Ruiyu> docker pull japeto/pujgcc
Using default tag: latest
latest: Pulling from japeto/pujgcc
aa18ad1a0d33: Pull complete
15a33158a136: Pull complete
f67323742a64: Pull complete
c4b45e832c38: Pull complete
e5d4afe2cf59: Pull complete
1efbd2d5674a: Pull complete
022a58c161f7: Pull complete
6461d3be7619: Pull complete
8287aab02fcd: Pull complete
edd3d1d6fad6: Pull complete
c6dbbe32db7e: Pull complete
```

Docker pull japeto/pujgcc:v0.12

```
PS C:\Users\lab> docker pull japeto/pujgcc:v0.12

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview japeto/pujgcc:v0.12
Error response from daemon: manifest for japeto/pujgcc:v0.12 not found: manifest unknown: manifest unknown
```

Docker pull japeto/pujgcc:latest

```
PS C:\Users\Ruiyu> docker pull japeto/pujgcc:latest
latest: Pulling from japeto/pujgcc
aa18ad1a0d33: Pull complete
15a33158a136: Pull complete
f67323742a64: Pull complete
c4b45e832c38: Pull complete
e5d4afe2cf59: Pull complete
1efbd2d5674a: Pull complete
022a58c161f7: Pull complete
6461d3be7619: Pull complete
8287aab02fcd: Pull complete
edd3d1d6fad6: Pull complete
c6dbbe32db7e: Pull complete
Digest: sha256:9b3d7bbf410396d0a26e6bcffbb54e4239736d5a23ad694b03d93c26f9fc6868
Status: Downloaded newer image for japeto/pujgcc:latest
docker.io/japeto/pujgcc:latest

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview japeto/pujgcc:latest
PS C:\Users\Ruiyu>
```

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

Esto se debe a que las imágenes de Docker están compuestas por múltiples capas, donde cada capa representa una parte de la imagen final. Cuando ejecutas el comando para descargar una imagen de Docker, el sistema descarga estas capas por separado. Cada mensaje de "Pull complete" indica que una capa específica se ha descargado con éxito.

Docker images

```
PS C:\Users\Ruiyu> docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
japeto/pujgcc       latest          05141310a94c   19 months ago  1.39GB
PS C:\Users\Ruiyu>
```

What is the size of the japeto/pujgcc image ?

El tamaño de la imagen es 1.39 GB

What are the options and parameters of the 'run' command ?

Comando docker run se utiliza para crear y ejecutar un contenedor a partir de una imagen especificada. Es uno de los comandos más utilizados en Docker y tiene numerosas opciones y parámetros para controlar el comportamiento del contenedor.

Entre estos:

- **-d**: Ejecutar el contenedor en segundo plano.
- **-p**: Mapear puertos del contenedor al host.
- **-e**: Establecer variables de entorno.
- **-v**: Montar volúmenes.
- **--name**: Asignar un nombre al contenedor.

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

Un contenedor permite manejar una imagen, esto quiere decir que la imagen es un template mientras que el contenedor es un ambiente aislado que ejecuta imágenes, y trabaja sobre ellas.

`docker run japeto/pujgcc:latest ls /data`

```
PS C:\Users\lab> docker run japeto/pujgcc:latest ls /data
ls: cannot access /data: No such file or directory
```

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

```
PS C:\Users\lab> docker run -v ${PWD}:/data/ japeto/pujgcc:latest ls /data/
AppData
Configuración local
Contacts
Cookies
DB2.sql
Datos de programa
Desktop
Documents
Downloads
Entorno de red
Favorites
Impresoras
Links
Menú Inicio
Mis documentos
Music
NTUSER.DAT
NTUSER.DAT{5c9c38db-9061-11ee-abe9-989096c51c3f}.TM.blf
NTUSER.DAT{5c9c38db-9061-11ee-abe9-989096c51c3f}.TMContainer000000000000000001.regtrans-ms
NTUSER.DAT{5c9c38db-9061-11ee-abe9-989096c51c3f}.TMContainer000000000000000002.regtrans-ms
OneDrive
Pictures
Plantillas
Reciente
Saved Games
Searches
SendTo
Videos
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini
```

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

Si son iguales, esto es porque la opción `-v ${PWD}:/data/` monta el directorio actual (`${PWD}`) del host en el directorio `/data/` del contenedor, haciendo que ambos directorios sean equivalentes en cuanto a su contenido.

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

```
PS C:\Users\lab> docker run -v ${PWD}:/data/ japeto/pujgcc:latest gcc /data/test.c
PS C:\Users\lab> docker run japeto/pujgcc

PS C:\Users\lab> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES
d95a1613e38f   japeto/pujgcc  "/bin/bash"             3 minutes ago Exited (127) 1 second ago           goofy_babbage
7cf5e0f81859   japeto/pujgcc  "/bin/bash"             7 minutes ago Exited (127) 6 minutes ago           awesome_hugle
578dbe3db3b3   japeto/pujgcc  "/bin/bash"             12 minutes ago Exited (0) 11 minutes ago           upbeat_dijkstra
329286516c1b   japeto/pujgcc  "/bin/bash"             41 minutes ago Exited (127) 38 minutes ago           upbeat_villani
149f1ce01e39   ubuntu         "/bin/bash"             41 minutes ago Exited (0) 41 minutes ago           kind_goldberg
341daaf61369   japeto/pujgcc  "/bin/bash"             44 minutes ago Exited (0) 42 minutes ago           pedantic_jang
3aace349c4ee   japeto/pujgcc  "bash --help"           2 hours ago   Exited (0) 45 minutes ago           heuristic_hypatia
PS C:\Users\lab> docker start awesome_hugle
awesome_hugle
```

Start container "docker start container".

```
PS C:\Users\lab> docker exec -it awesome_hugle /bin/bash
root@7cf5e0f81859:/# exit
exit
```

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

No, el contenedor sigue corriendo aun si ponemos el comando `exit`.

```
docker ps -l
```

```
PS C:\Users\lab> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES
7cf5e0f81859   japeto/pujgcc  "/bin/bash"             18 minutes ago Up 10 minutes           awesome_hugle
```

```
docker stop mycontainer
```

```
PS C:\Users\lab> docker stop awesome_hugle
awesome_hugle
PS C:\Users\lab> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES
```