

COMPUTACIÓN TOLERANTE A FALLAS

Ingeniería en computación

López Franco Michel Emanuel

CUCEI



CORREO: juan.guillen9059@alumnos.udg.mx

ALUMNO: Guillén García Juan Andrés

TELEFONO: +52 33 3821 0156

CÓDIGO: 220790598

SECCIÓN: D06

CICLO: 24A

Fecha: 25 / Febrero / 2024

Introducción

¿Qué es un hilo?

Es una unidad básica de utilización de CPU, la cual contiene un id de hilo, su propio program counter, un conjunto de registros, y una pila; que se representa a nivel del sistema operativo con una estructura llamada TCB (thread control block). Los hilos comparten con otros hilos que pertenecen al mismo proceso la sección de código, la sección de datos, entre otras cosas. Si un proceso tiene múltiples hilos, puede realizar más de una tarea a la vez (esto es real cuando se posee más de un CPU).

Ventajas:

- Respuesta: el tiempo de respuesta mejora, ya que el programa puede continuar ejecutándose, aunque parte de él esté bloqueado.
- Compartir recursos: los hilos comparten la memoria y los recursos del proceso al que pertenecen, por lo que se puede tener varios hilos de ejecución dentro del mismo espacio de direcciones.
- Utilización múltiples CPUs: permite que hilos de un mismo proceso ejecuten en diferentes CPUs a la vez. En un proceso mono-hilo, un proceso ejecuta en una única CPU, independientemente de cuantas tenga disponibles.

¿Para que sirve un Demonio en programación?

Son procesos que se ejecutan continuamente en segundo plano y realizan funciones necesarias para otros procesos. TCP/IP (Transmission Control Protocol/Internet Protocol) proporciona daemons para implementar determinadas funciones en el sistema operativo. Estos daemons son procesos en segundo plano que se ejecutan sin interrumpir otros procesos (a menos que esto forme parte de la función del daemon).

¿Qué permite la programación concurrente?

La programación concurrente permite desarrollar software que ejecuta eventos o circunstancias que están sucediendo o existen al mismo tiempo. Los módulos concurrentes interactúan enviándose mensajes entre sí.

¿Qué es un servicio?

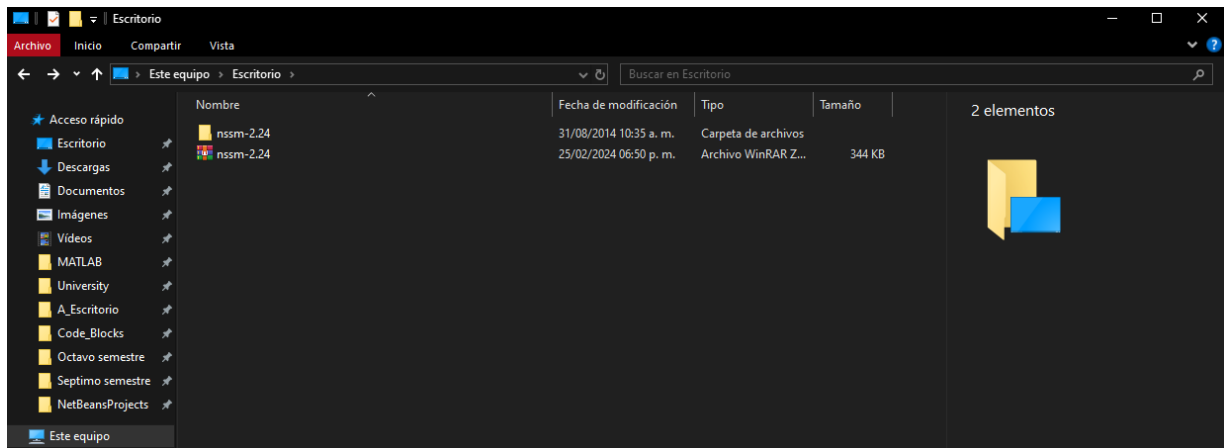
Es un programa o proceso que se ejecuta en segundo plano en un sistema operativo sin necesidad de interacción directa por parte del usuario. Los servicios suelen realizar tareas específicas o proporcionan funcionalidades para el sistema operativo o para otras aplicaciones.

Objetivo

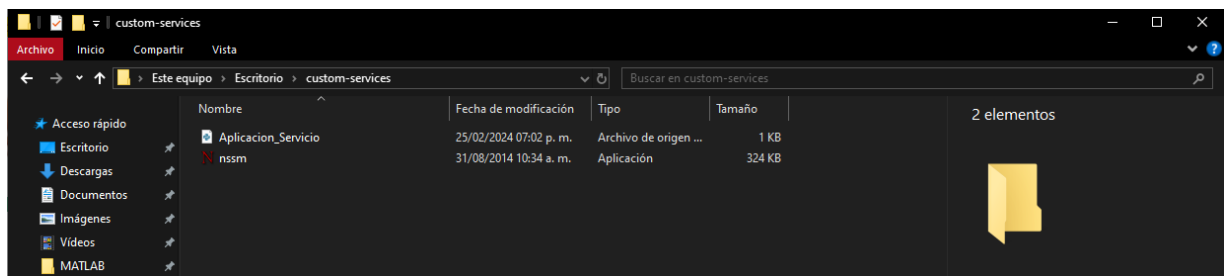
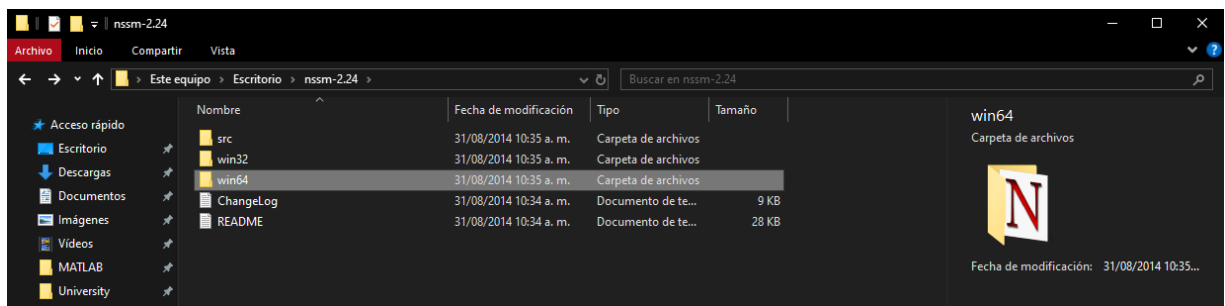
Realizar un programa en el cual permita el poder ver el estado del programa o aplicación en cuestión.

Desarrollo

Primero descargamos el software Non-Sucking Service Manager (NSSM) desde el siguiente enlace "<https://nssm.cc/download>", para a continuación descomprimirlo y crear una nueva carpeta que se llama "custom-services".

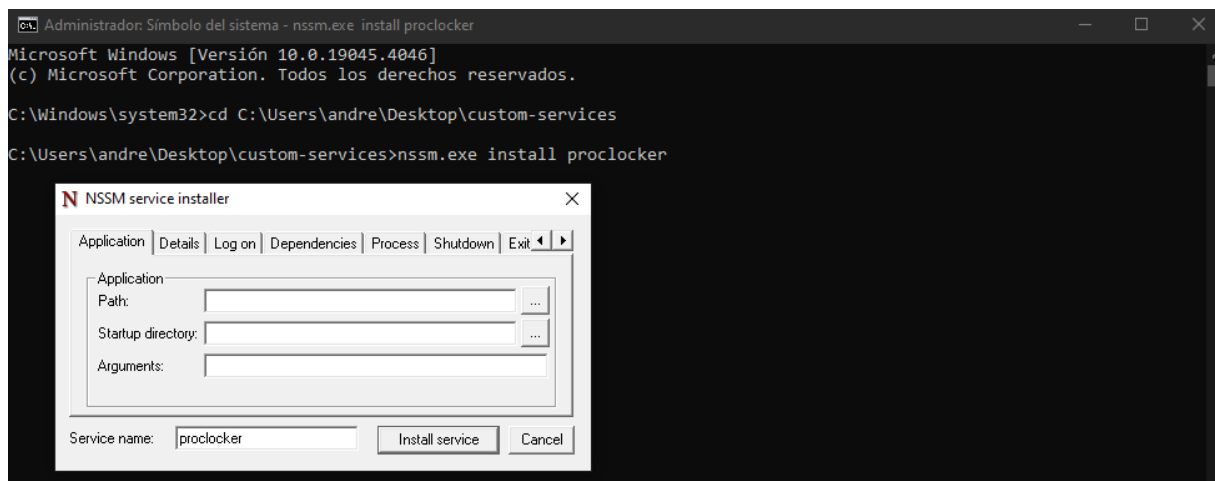


Dentro de dicha carpeta guardaremos el programa en Python, también buscaremos dentro de la carpeta descomprimida "nssm-2.24", en esta parte abriremos la carpeta "win64" y el ejecutable del programa lo moveremos a la carpeta "custom-sevices" que creamos previamente en donde también esta guardado el programa creado en Python.

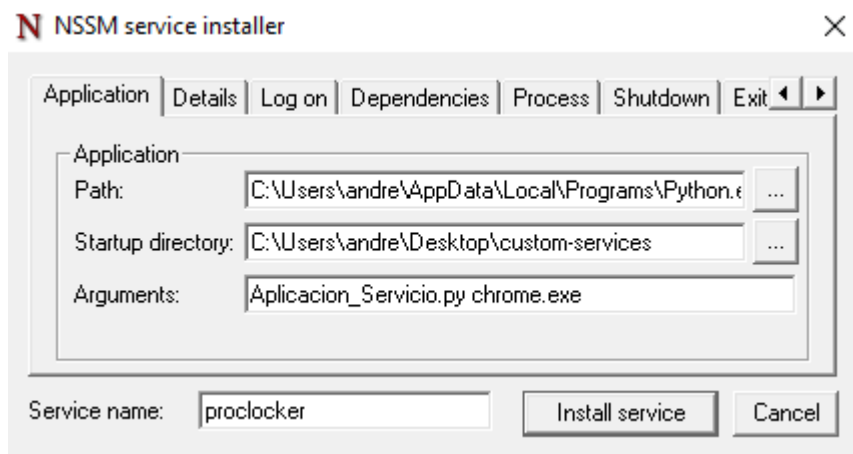


Estatus

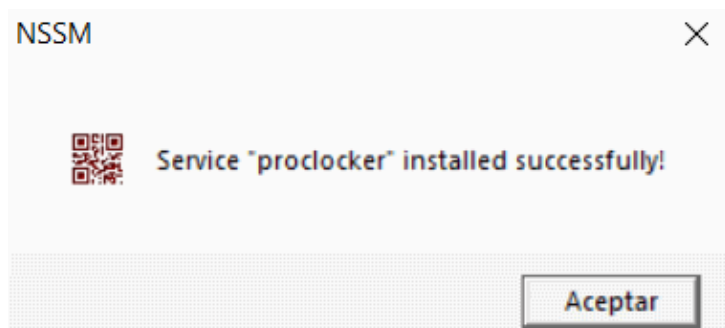
Ahora para iniciar el servicio abriremos el CMD como administrador y pondremos la dirección de la carpeta “custom-services”, para a continuación instalar por medio de la siguiente línea “nssm.exe install proclocker” el software.



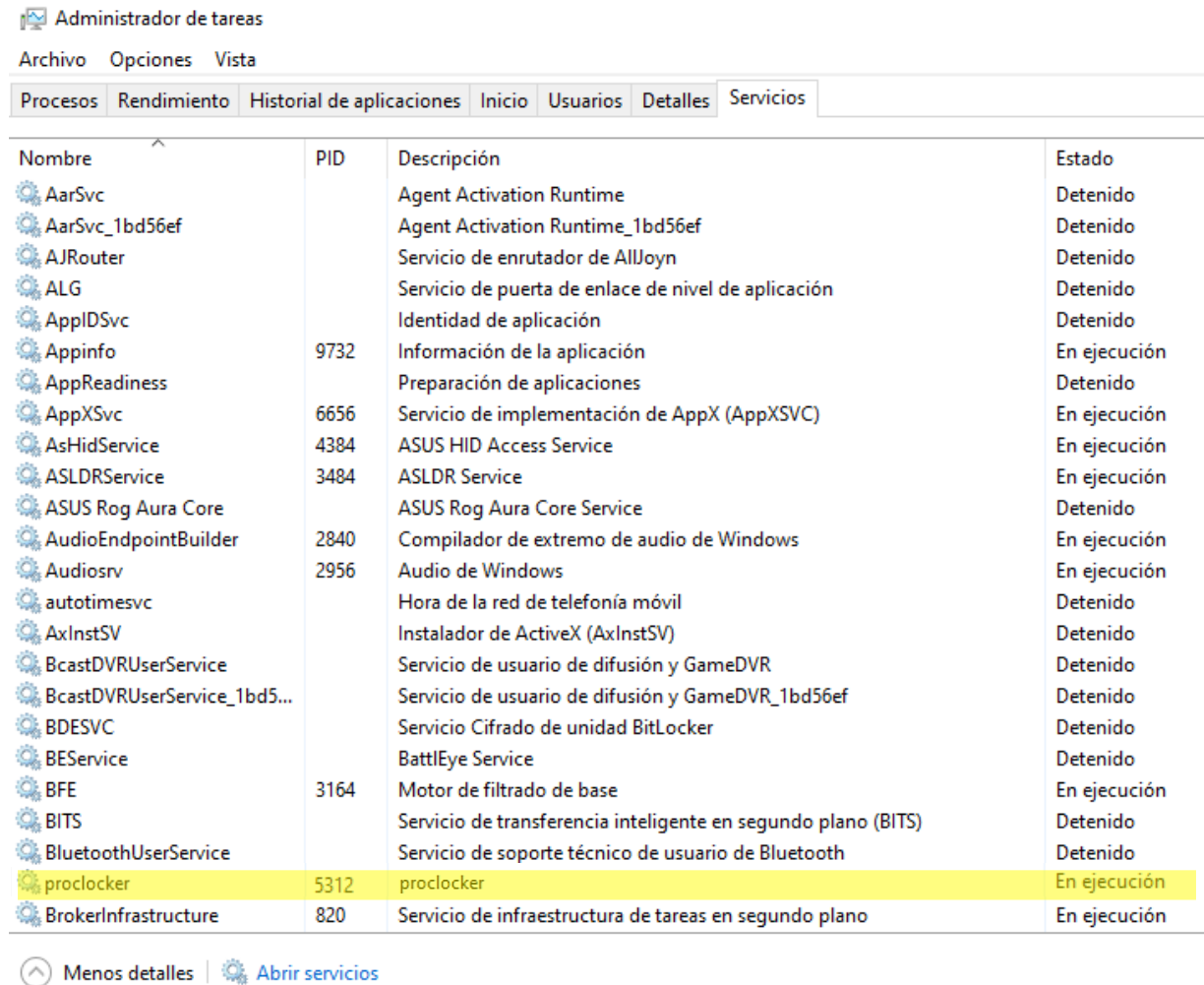
La línea anterior para la instalación del servicio nos abre la ventana que se puede observar en la siguiente imagen y en la anterior. En dicha ventana rellenaremos los espacios necesarios con la ubicación del path, la carpeta “custom-services” y en los argumentos el nombre del programa junto con el buscador al cual le queremos aplicar el servicio, en mi caso el navegador de Chrome.



Una vez rellenado los campos, presionamos “Install service”. En donde nos saldrá otra ventana diciendo que el servicio ha sido instalado.



Luego de haber puesto en el CMD la línea de comando “C:\Users\andre\Desktop\custom-services>nssm.exe star proclocker”, sale un mensaje en donde dice que la operación se completó correctamente. Como se puede ver en la siguiente imagen muestro el administrador de tareas y llendome a la opción de servicios se encuentra en la parte inferior el servicio de proclocker en ejecución.



Y haciendo uso del siguiente comando “C:\Users\andre\Desktop\custom-services>nssm.exe stop proclocker” detenemos el servicio, donde nos muestra un mensaje confirmando la acción realizada.

```
C:\custom-services>nssm.exe stop proclocker
proclocker: STOP: La operación se completó correctamente.
```

Estatus

En la siguiente imagen se muestra como al querer detenerlo de nuevo o hacer cualquier acción con el servicio se muestra un mensaje que nos dice que el servicio no puede abrirse y que no está instalado. Esto sucede porque previamente use el comando “nssm.exe remove proclocker” el cual borra el servicio.

A screenshot of a Windows command prompt window titled "Administrador: Símbolo del sistema". The window shows the following text: "Microsoft Windows [Versión 10.0.19045.4046] (c) Microsoft Corporation. Todos los derechos reservados. C:\Windows\system32>cd C:\Users\andre\Desktop\custom-services C:\Users\andre\Desktop\custom-services>nssm.exe stop proclocker Can't open service! OpenService(): El servicio especificado no existe como servicio instalado."

Con la anterior imagen demuestro el como si se detuvo el servicio y ya no aparece en la terminal, junto con sus respectivos mensajes de error.

Código de python

En la primera función del programa se encarga de evitar errores en el programa, así mandando información y por último cerrándose con “sys.exit(0)”.

```
import sys
import psutil

def check_arguments():
    if len(sys.argv) == 1:
        print('Este programa no funciona sin argumentos')
        sys.exit(0)
```

Aquí busca algún objetivo ejecutable, donde se encarga de verificar si existe o no.

```
def get_targets():
    targets = sys.argv[1:]
    i = 0
    while i < len(targets):
        if not targets[i].endswith('.exe'):
            targets[i] = targets[i] + '.exe'
            i += 1
    return targets
```

Cierra el proceso que se ejecute junto con los argumentos previamente recibidos, esto básicamente tiene la función que le asigne por medio de “nssm” que cada vez que abra Google Chrome se cierre inmediatamente una vez que se abra.

```
def lock(target):
    for proc in psutil.process_iter():
        if proc.name().lower() == target.lower():
```

```
proc.kill()

if __name__ == '__main__':

    check_arguments()
    targets = get_targets()

    while True:
        for target in targets:
            lock(target)
```

Conclusión

El hacer esta práctica me resulto un poco enrevesada por los motivos de que no entendí bien lo que se solicitaba, así que en base a la información y los pasos que se mostraban en unos de los enlaces de la actividad, fue como estuve desarrollando y dirigiendo esta práctica. Al final por la parte de iniciar servicios me resulto bastante entretenida porque cuando abría Google Chrome se me cerraba casi instantáneamente mientras tenía el servicio encendido en mi computadora, incluso al apagar y prender la computadora sin haber detenido el servicio, este seguía aplicándose de la misma manera y así seguía cerrándome el navegador hasta que lo apagué y removí desde la consola.

Estatus

Bibliografía

estarache @gmail.com, tecnobillo. (2022, March 12). Crear Servicios para Windows con Python. Tecnobillo.com. <https://tecnobillo.com/sections/python-en-windows/servicios-windows-python/servicios-windows-python.html>

NSSM - the Non-Sucking Service Manager. (n.d.). Nssm.cc. Retrieved February 26, 2024, from <https://nssm.cc/download>

threading — Paralelismo basado en hilos — documentación de Python - 3.8.18. (n.d.). Docs.python.org. Retrieved February 19, 2024, from <https://docs.python.org/es/3.8/library/threading.html>