

Universidad de Guadalajara.

Computación Tolerante a fallas.



Diego Ivan Becerra Gonzalez.

Sección D06.

## Objetivo.

Realizar un programa que sea capaz de revisar el estatus de tu aplicación.

## Desarrollo.

Para comenzar, la diferencia entre un proceso y un servicio o demonio es que el proceso necesita la intervención del usuario y tienen un interfaz gráfica, mientras que el servicio no, puede ser ejecutado sin necesidad de intervención del usuario, por ejemplo, cuando el equipo enciende.

Esta actividad se realizó siguiendo las instrucciones en el link publicado.

Primero, creamos un programa en Python el cual recibe como parámetros nombres de programas. Lo que hace este script es revisar que los programas mandados como parámetros no estén activos, y en caso de estarlo, los cierra. Esto se realiza dentro de un bucle while infinito, para que lo haga continuamente.

```
1  import sys
2  import psutil
3
4  def check_arguments():
5      if len(sys.argv) == 1:
6          print('Este programa no funciona sin argumentos')
7          sys.exit(0)
8
9  def get_targets():
10     targets = sys.argv[1:]
11     i = 0
12     while i < len(targets):
13
14         if not targets[i].endswith('.exe'):
15             targets[i] = targets[i] + '.exe'
16             i += 1
17     return targets
18
19 def lock(target):
20     for proc in psutil.process_iter():
21         if proc.name().lower() == target.lower():
22             proc.kill()
```

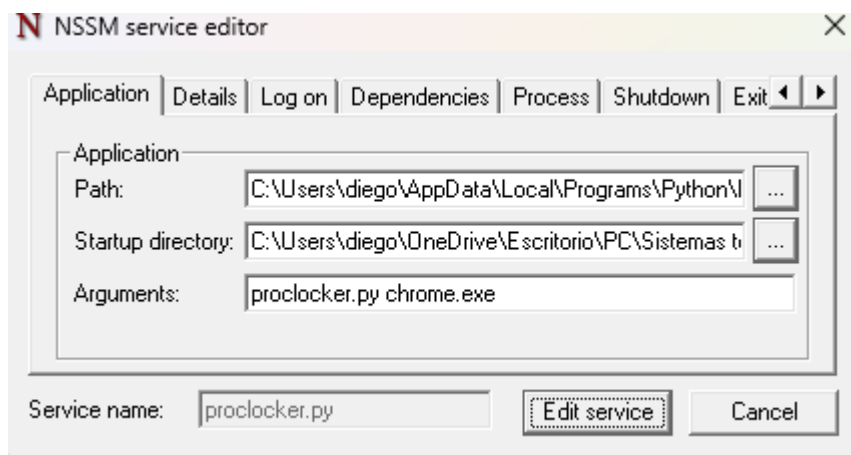
```

if __name__ == '__main__':
    check_arguments()
    targets = get_targets()
    while True:
        for target in targets:
            lock(target)

```

Después, descargamos una herramienta llamada the non-sucking service manager, la cual nos permite crear servicios en Windows. Copiamos el programa en la carpeta donde tenemos nuestro script y la instalamos desde la consola con el siguiente comando: `nssm.exe install [nombre del script]`.

Eso nos abrirá un recuadro como el que sigue:



Ahí escribimos los siguientes parámetros: la ruta donde tenemos Python, la ruta de nuestro script y el nombre del script con sus parámetros, en este caso, vamos a probarlo con Chrome.

Una vez creado el servicio, lo iniciamos con `nssm.exe start [servicio]`.

Y lo que hace es que, si teníamos ventanas de Chrome abiertas, las cierra, y si queremos abrir más, no nos deja, esto porque se ejecuta continuamente.

Para parar el servicio cambiamos la palabra start por stop, y para desinstalarlo, cambiamos start por remove.

Cabe mencionar que antes de desinstalarlo se inició el servicio, se reinicio el equipo y el servicio seguía corriendo, sin necesidad de hacer todos los pasos anteriores.

### Conclusión.

La creación de servicios nos es útil cuando necesitamos asegurarnos de que un programa se ejecute constantemente, sin la necesidad de que algún usuario lo ejecute de manera voluntaria, ejemplo de estos programas serian servidores web o alguna herramienta de diagnóstico del equipo.