Universidad de Guadalajara

Ingeniería en computación

Computación Tolerante a Fallas 2023ª

“Estatus”

Olguín Hernández Jair Benjamín.

217439707

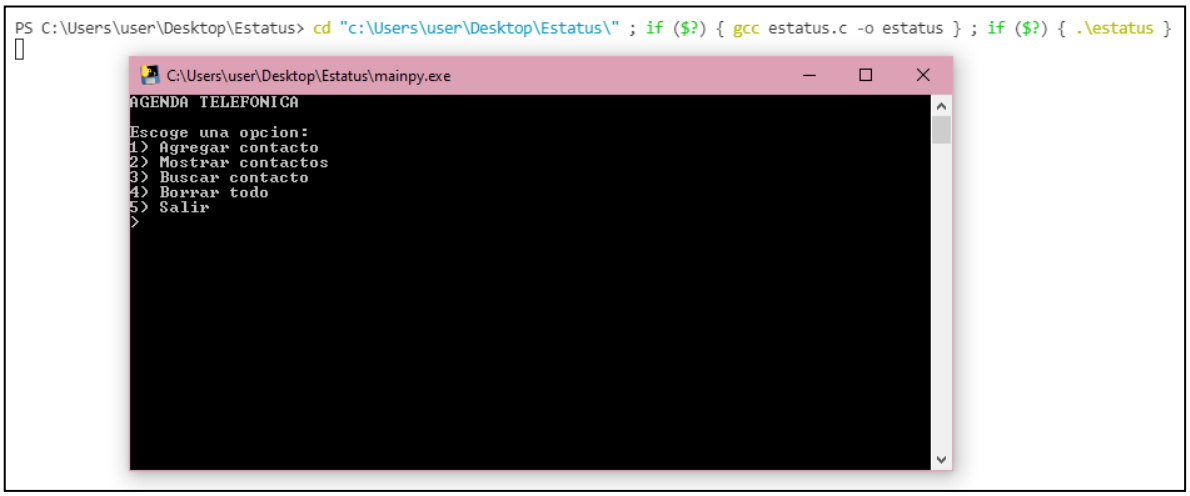
Para poder revisar el estatus de mi programa de Python de la agenda telefónica cree un programa en lenguaje C que es capaz de revisar todos los procesos que están activos. Para ello utilicé la función popen(), la cual permite llamar a un comando de Shell y obtener su resultado de salida en un archivo. El comando que utilicé es el de “tasklist” que es para visualizar los procesos que están ejecutándose en el sistema.

```
FILE *p = popen("tasklist", "r");
```

Después con una iteración que va leyendo cada línea del archivo (o sea cada proceso activo) hasta que se termine, se verifica si es que se encuentra “mainpy.exe”, que es un archivo ejecutable del programa en Python de la agenda que generé con el modulo Pyinstaller para así poder identificarlo más fácilmente en el tasklist. Si no se encontró el proceso entonces lo ejecuta con system() que sirve para ejecutar comandos del sistema. Todo esto se repite en un bucle infinito para estar revisando que la aplicación no se cierre, y si por alguna razón se cierra, volverla a ejecutar.

```
while (fgets(linea, sizeof(linea), p)) {
    if(strstr(linea,"mainpy.exe")) {
        encontrado = true;
    }
}
if (encontrado == false)
    system("start mainpy.exe");
```

Al ejecutar el programa en C (estatus.c) se ejecuta también el .exe de la agenda en Python ya que no se estaba ejecutando, y si se cierra la ventana del ejecutable, esta se vuelve a abrir de inmediato:



Si se están introduciendo datos de un contacto en la aplicación y estos no se terminan de introducir y no se guarda el contacto, y de pronto la aplicación se cierra como ocurre aquí:

```
Escoge una opcion:
1) Agregar contacto
2) Mostrar contactos
3) Buscar contacto
4) Borrar todo
5) Salir
> 1
Introduce el nombre del contacto: Jair Olguin
Introduce su telefono: 1122334455
Introduce su correo:
```

Al volverse a abrir la aplicación (gracias a estatus.c) y querer volver a introducir un contacto,los datos que ya se habían escrito antes de que la aplicación se cerrara se restauran y se puede continuar escribiendo los datos del contacto:

```
Escoge una opcion:
1) Agregar contacto
2) Mostrar contactos
3) Buscar contacto
4) Borrar todo
5) Salir
> 1
Introduce el nombre del contacto: Jair Olguin
Introduce su telefono: 1122334455
Introduce su correo:
```

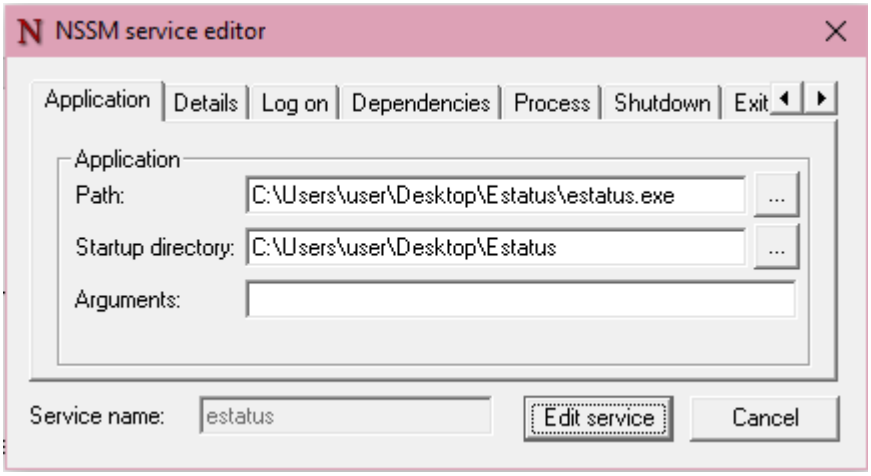
Esto gracias a un hilo que se está ejecutando cada vez que se introduce un contacto, y cada vez que el usuario guarda un dato del contacto que quiere agregar se abre un archivo de respaldo pickle y se guarda el diccionario con el dato (o los datos) que se tengan hasta ese momento para que así se puedan restaurar cuando se vuelva a abrir la aplicación.

```python
hilo1 = threading.Thread(target=revisar, args=(contacto,))
hilo1.start()
```

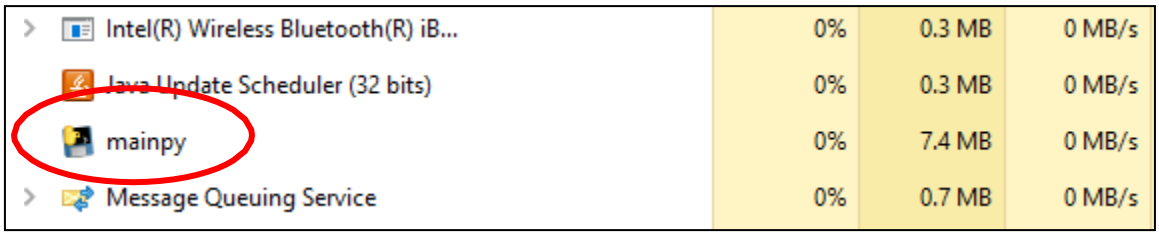
```python
def revisar(dato):
    global guardar
    while bandera == True:
        if guardar == True:
            respaldo = open('respaldo.pickle', 'wb')
            pickle.dump(dato, respaldo)
            respaldo.close()
            guardar = False
```

Si sí se termina de agregar el contacto y se guarda en el archivo de datos.pickle (contactos agregados), el archivo de respaldo.pickle se borra, y se vuelve a crear al agregar otro contacto ya que se vuelve a lanzar el hilo.

El programa de estatus.c que es el que revisa que la aplicación esté funcionando puede convertirse en un servicio para que siempre se inicie al reiniciar el sistema con Non-Sucking Service Manager (NSSM) y así verificar que la aplicación de la agenda siempre esté ejecutandose. Para ello se tiene que instalar el nssm.exe previamente descargado y copiado en la carpeta de trabajo, y se abre el formulario a llenar con los datos de estatus.c



Una vez instalado el servicio, se debe iniciar desde la consola con “nssm.exe start estatus”, y ya podrá verse que el archivo ejecutable de la aplicación de la agenda se está ejecutando en segundo plano, y si se finaliza la tarea, inmediatamente esta se vuelve a ejecutar:



Conclusión:

Esta actividad me pareció muy interesante ya que comprendí la diferencia que hay entre una aplicación y un servicio. Otra cosa que desconocía era el funcionamiento de los hilos, pues pueden ser muy útiles si se quieren ejecutar varias operaciones en el mismo espacio de proceso, lo cual resulta más eficiente.

Bibliografía:

Commands CMD to c++ - C++ Forum. (2020, 11 abril). cplusplus. Recuperado 20 de febrero de 2023, de http://www.cplusplus.com/forum/beginner/269490/

Funciones raras, pero útiles, en C de unix/linux. (2007, 4 febrero). lsi.vc.ehu.eus. Recuperado 20 de febrero de 2023, de https://lsi.vc.ehu.eus/pablogn/docencia/manuales/SO/funciones.php.html

Ejecutables Python – Python 3 para todo. (2020, 2 junio). Python 3 para todo. Recuperado 20 de febrero de 2023, de https://www.pythonparatodo.com/?p=74

Python 3 para impacientes. (2016, 16 diciembre). python-para-impacientes. Recuperado 20 de febrero de 2023, de https://python-para-impacientes.blogspot.com/2016/12/threading- programacion-con-hilos-i.html