

Activitat processos de Linux

Para hacer este proyecto utilice matplotlib para generar los graficos.

el proyecto esta combinado con python y bash, porque utilizo un subprocesso para pasar toda la informacion de Linux al terminal de python. en uno de los casos para llamar al proceso 'pstree' utilice con el modulo 'subprocess' que pasa toda la informacion como un conjunto de bytes, de esa misma forma hay que capturar toda informacion para tratarla, en este caso queria sacar toda esta informacion por la salida estandar por lo que tuve que decodificar toda la informacion tratada por 'subprocess'

```
def opcion1():  
  
    process = subprocess.run(  
        ['pstree'], capture_output=True)  
  
    print(process.stdout.decode())
```

opcion dos: para generar el grafico cree a proposito un script que solo captura los primeros 10 procesos del sistema, con su PID y nombre. y con el modulo networkx as nx puedo trabajar en conjunto con matplotlib para editar la manera en la que quiero mostrar mi grafico.

En esta opcion utilice otro modulo que es psutil (muy parecido a subprocess) pero por que utilice este psutil y no subprocess en esta opcion? basicamente porque la API de psutil es mas sencilla para procesar las metricas del sistema, lo que lo hace mas facil para concatenar todas estas metricas con matplotlib

```

def option2():
    G = nx.DiGraph()
    process_count = 0
    max_process_count = 10

    for process in psutil.process_iter(['pid', 'ppid', 'name']):
        if process_count >= max_process_count:
            break

        pid = process.info['pid']
        ppid = process.info['ppid']
        G.add_node(pid, label=f"{process.info['name']} (PID: {pid})")
        if ppid != 0:
            G.add_edge(ppid, pid)
        process_count += 1

    pos = nx.spring_layout(G)

    label_names = nx.get_node_attributes(G, 'label')
    nx.draw_networkx_nodes(G, pos, node_size=500)
    nx.draw_networkx_edges(G, pos, edgelist=G.edges(), edge_color='black')
    nx.draw_networkx_labels(G, pos, labels=label_names, font_size=10)

    plt.title("Map of processes with their PID")
    plt.show()

```

en la opcion numero 3 reutilice la misma funcion de la opcion dos, solo para agregarle colores a los nodos. aqui utilice tambien otra libreria que se llama **numpy as np** para trata los procesos como arrays y asi asignarle un color a cada array

DISCLAIMER: NO ENTENDI MUY BIEN EL ENUNCIADO Y NO SABIA SI HABIA QUE AGREGARLE UNA OPCION PARA QUE EL USUARIO ELIJA QUE PROCESO RESALTAR, POR LO QUE DECIDI RESALTAR TODOS LOS PROCESOS DE TODOS LOS COLORES

En la opcion 4. utilice el subprocesso para capturar la informacion de los procesos que se estan ejecutando en el sistema

REFERENCIAS:

<https://www.devgem.io/posts/comparing-python-graph-libraries-networkx-igraph-graph-tool-and-python-graph>

<https://cienciadedatos.net/documentos/pygml03-analisis-redes-python-networkx>

<https://realpython.com/tutorials/numpy/>

https://networkx.org/documentation/stable/reference/generated/networkx.classes.function.get_node_attributes.html

<https://networkx.org/documentation/stable/index.html>