

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Sistemas Operativos 2
Sección A



Manual Tecnico

GRUPO 4

Repositorio	https://github.com/Yaiza81/SO2_1S2022_P2_G4
-------------	---

Manual Técnico:

Introducción:

En el siguiente escrito se tiene la finalidad de proporcionar al usuario la lógica utilizada para realizar la aplicación de la práctica 2 y con la cual se desarrollan las funcionalidades que posee, además de documentar los pasos empleados en la construcción del programa.

Este manual no está dedicado para enseñar a cómo utilizar la aplicación, ni las herramientas utilizadas para el desarrollo del programa. Para conocer la guía de la correcta utilización del programa se debe consultar el “Manual de Usuario”.

Objetivo del manual:

Se pretende dar a conocer las herramientas y la manera en que se utilizaron para programar la siguiente aplicación, además de la lógica empleada para las características extra del programa.

Aplicación:

Requerimientos técnicos:

Requerimientos mínimos de hardware:

- ☐ Equipo físico con teclado, mouse y monitor
- ☐ Procesador (core i3 en adelante)
- ☐ Memoria RAM de 1 GB
- ☐ Disco duro con al menos 50 GB de espacio libre
- ☐ Tarjeta de red LAN y/o Wireless

Requerimientos mínimos de software:

- ☐ Sistema operativo Windows o Ubuntu

Descripción de herramientas utilizadas:

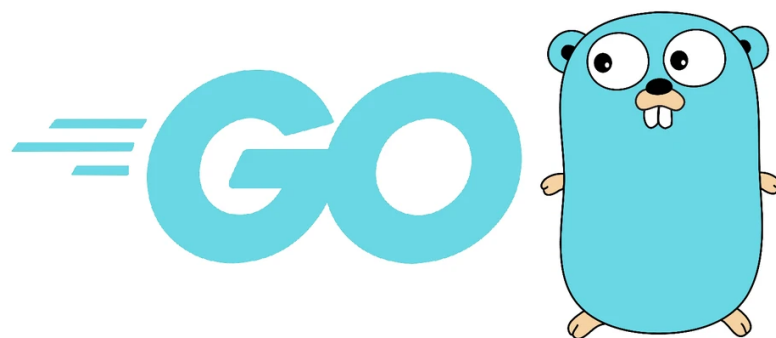
➤ Visual Studio Code

Es un IDE diseñado para programar, depurar, probar e implementar soluciones en múltiples plataformas. Posee varias características novedosas que permiten que el desarrollo de aplicaciones sea más optimizado y mejor. Desde herramientas que permiten autocompletado inteligente, integraciones con software de control de versiones, integración con la nube, herramientas de debug, integraciones con gestores de base de datos, entre muchas otras. Resalta por ser bastante personalizable pudiendo cada usuario añadir las características que desea, editar los atajos necesarios y preferencias cualquiera. Es gratuito y de código abierto.



➤ Lenguajes de programación: Golang

Golang es un lenguaje de programación de código abierto creado por Google. Su enfoque principal es la productividad, algo esencial para quien actúa con sistemas multiplataforma. Popularmente conocido como Go, el lenguaje se diferencia de los demás por ser compilado y altamente escalable..



➤ Control de versiones: GitHub

GitHub es una forja para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador.

Es una plataforma de gestión y organización de proyectos basada en la nube que incorpora las funciones de control de versiones de Git. Es decir que todos los usuarios de GitHub pueden rastrear y gestionar los cambios que se realizan en el código fuente en tiempo real, a la vez que tienen acceso a todas las demás funciones de Git disponibles en el mismo lugar. Además, la interfaz de usuario de GitHub es más fácil de usar que la de Git, lo que la hace accesible para personas con pocos o ningún conocimiento técnico.



Librerías principales:

- Colly

Librería que provee de una interfaz para la construcción de un scrapper. Permite extraer fácilmente datos estructurados de páginas web, que pueden ser utilizados para diversas aplicaciones, como minería de datos, procesamiento de datos o archivos.

- BubbleTea

Librería que permite la creación de aplicaciones de terminal, permite desarrollar la consola y adaptarla a las necesidades del program que se está codificando y así mostrarlo con mayor detalle.

- crypto/sha256

Este paquete implementa los algoritmos de Hash SHA224 y SHA256 definidos en el FIPS 180-4

- encoding/json

Le permite al programa implementar funciones de mapeo y desmapeo entre JSON y valores de GO para las funciones de Marshall y Unmarshall

Descripción de la aplicación:

- Uso de hilos

Se utilizaron dos hilos principales que fueron “Jobs” y “Results”, los cuales son utilizados por los monos para registrar la información que se les envía. De esta forma cada mono tiene un hilo en el cual podrá trabajar al mismo tiempo que los otros monos.

- Recursos compartidos

Primeramente se declara un struct “cache” el cual tiene en su interior una variable tipo ReadWrite Mutex que permitirá que diversas rutinas puedan escribir sobre ella. Se tiene la cola de espera creada a partir del struct “cache”. Se le declararon los métodos de “agregar”, “quitar” y “leer”, los cuales serán utilizados por los distintos procesos. Para evitar que un recurso trate de acceder a la cola mientras otro lo está usando se coloca la función “Lock()” antes de realizar las asignaciones a la cola y “Unlock()” una vez se termine la asignación para que se libere el recurso entre ambas sentencias.

- Comunicación y sincronización entre procesos

La comunicación y sincronización de los procesos se lleva a cabo por los workers (monos), los cuales ejecutan diversas sentencias cuyos resultados deben ser transmitidos a otros monos o bien almacenarse. Para la transmisión de datos utilizan los canales para enviar los recursos con los que están trabajando y que los demás monos puedan utilizar dichos recursos. Una vez un mono ha terminado sus datos envía una respuesta a través del canal “ResponseMsg” el cual permite que los procesos se comuniquen y transmitan información entre ellos. Por medio del canal “jobs” van efectuando las tareas de scrapping y recolección de datos en base a los datos que le comunique su antecesor. Y ellos le comunican a su predecesor los resultados de sus tareas a través del canal “results”.

Lógica implementada en la aplicación:

- PROCESOS E HILOS

Los procesos son una abstracción o instancia de un programa en ejecución.

Es cuando las instrucciones de un programa (líneas de código) son ejecutadas por el procesador.

Un proceso es un programa ejecutándose dentro de su propio espacio de direcciones”.

Se puede decir que un proceso es un supervisor de hilo(s) de ejecución. Un hilo es una secuencia de código en ejecución dentro del contexto de un proceso. Los hilos no pueden ejecutarse ellos solos

- MULTIPROGRAMACIÓN

La multiprogramación es una técnica de multiplexación que permite la ejecución simultánea de múltiples procesos en un único procesador. En realidad, esta técnica produce una ilusión de paralelismo, de manera que parece que todos los procesos se están ejecutando a la vez

- PARALELISMO

El Paralelismo en la informática , es una función que realiza el procesador para ejecutar varias tareas al mismo tiempo. Es decir, puede realizar varios cálculos simultáneamente, basado en el principio de dividir los problemas grandes para obtener varios problemas pequeños, que son posteriormente solucionados en paralelo.

- CONCURRENCIA

En ciencias de la computación, concurrencia se refiere a la habilidad de distintas partes de un programa, algoritmo, o problema de ser ejecutado en desorden o en orden parcial, sin afectar el resultado final.

Los cálculos (operaciones) pueden ser ejecutados en múltiples procesadores, o ejecutados en procesadores separados física o virtualmente en distintos hilos de ejecución. Un sin número de modelos matemáticos han sido desarrollados para cálculos de la concurrencia en general incluyendo redes de Petri, procesos Calculi, el modelo máquina de accesos random en paralelo, el Modelo Actor y el Lenguaje Reo

- COMUNICACIÓN ENTRE PROCESOS

La comunicación entre procesos (comúnmente IPC, del inglés Inter-Process Communication) es una función básica de los sistemas operativos. Los procesos pueden comunicarse entre sí a través de compartir espacios de memoria, ya sean variables compartidas o buffers, o a través de las herramientas provistas por las rutinas de IPC. La IPC provee un mecanismo que permite a los procesos comunicarse y sincronizarse entre sí, normalmente a través de un sistema de bajo nivel de paso de mensajes que ofrece la red subyacente.