

---

## Universidad de San Carlos de Guatemala

Facultad de Ingeniería

1er. Semestre del 2022

Sistemas Operativos 2. Sección "A"

Ing. César Batz Saquimux

Aux. Orfrant Rivera

Aux. Pablo Barillas

# Práctica #2

## Programación concurrente.

### OBJETIVOS

1. Poner en práctica los conocimientos adquiridos sobre procesos e hilos.
2. Resolver problemas que requieren técnicas de programación concurrente.
3. Desarrollar software utilizando de forma eficiente múltiples hilos de ejecución y los recursos del sistema operativo.
4. Coordinar múltiples procesos y tareas en programas utilizando los conocimientos adquiridos sobre programación concurrente para la correcta comunicación y sincronización entre procesos.

### INSTRUCCIONES GENERALES

En los grupos establecidos del laboratorio, se les solicita a los estudiantes desarrollar las siguientes aplicaciones y resolver el siguiente problema utilizando el lenguaje de programación Golang; y, sobre todo, aplicando de forma correcta los conocimientos vistos en el laboratorio sobre programación concurrente y desarrollo de software con múltiples hilos.

Para cada problema se debe realizar un análisis de la situación, en el análisis deberán identificar las situaciones en las cuáles existen múltiples procesos, las variables o datos que comparten los procesos, la forma en que se sincronizan y las situaciones que pueden llevar a problemas como: condiciones de carrera o deadlocks. Además, se debe indicar cómo se solucionan dichos problemas y qué herramientas y conceptos se implementaron, todo esto deberá ir en la documentación del proyecto y será de gran valor e importancia, ya que es una forma de evaluar que el estudiante tiene conocimiento del software que implementó. En concreto, se evaluará que cada uno de los manuales técnicos describen:

- Uso de hilos.
- Recursos compartidos.

- 
- Comunicación y sincronización entre procesos.

## ESPECIFICACIONES

### Los monos buscadores

El problema de los monos buscadores, basado en el problema de los barberos dormilones de Dijkstra. Se debe realizar una aplicación que hará funciones de un web scraper de contenido de wikipedia en el cual se simulará el trabajo de monos buscadores, los cuales estarán encargados de realizar las búsquedas dentro de wikipedia.

Al iniciar el programa se deben configurar los siguientes parámetros:

1. Cantidad de monos buscadores.
2. Tamaño de la cola de espera.
3. Un número  $Nr$  que representa las  $n$  referencias más mencionadas en el artículo.
4. Búsqueda inicial o url inicial.
5. Nombre del archivo donde se almacenará el resultado de la búsqueda.

Los monos deben realizar el siguiente trabajo con cada artículo que leen, (solo se toma en cuenta el contenido dentro de la etiqueta `<p>` de html):

1. Contar la cantidad de palabras dentro del artículo de wikipedia.
2. Contar la cantidad de enlaces a otros artículos de wikipedia.
3. Obtener el sha-256 del contenido
4. Para la primera búsqueda se obtienen las  $n$  referencias más mencionadas y se agregan a la cola de espera haciendo que  $Nr = Nr - 1$  para esa nueva búsqueda, para que otro mono buscador las analice.
5. Si la cola está llena las referencias son descartadas.
6. El mono debe colocar su id al resultado de la búsqueda y guardarlo en el archivo.
7. El mono debe descansar un tiempo proporcional al número de palabras leídas antes de empezar otra búsqueda.

---

El archivo de resultado de la búsqueda debe ser escrito de forma concurrente por todos los monos conforme van terminando su proceso, la estructura esperada es un archivo json con un formato similar al siguiente:

```
1  [{
2      "origen": 0,
3      "conteo_palabras": 1000,
4      "conteo_enlaces": 20,
5      "sha": "23415kj23145kj2134k5j",
6      "url": "https://es.wikipedia.org/wiki/Chuck\_Norris",
7      "mono": "id1"
8  },
9  {
10     "origen": "23415kj23145kj2134k5j",
11     "conteo_palabras": 800,
12     "conteo_enlaces": 15,
13     "sha": "asddffdfasdfasdfa",
14     "url": "https://es.wikipedia.org/wiki/Ryan\_\(Oklahoma\)",
15     "mono": "id2"
16  }]
17
```

## CONSIDERACIONES

- La representación de la aplicación podrá ser a través de la consola.
- Se puede realizar un menú para configurar la cantidad de monos, la cantidad de búsquedas en espera y la url de la búsqueda inicial
- Cualquier copia total o parcial será reportada a la Escuela de Sistemas para que proceda como corresponde.
- La práctica será realizada por 4 personas utilizando los grupos establecidos en el laboratorio.

## ENTREGABLES

- Código fuente de las aplicaciones.
- Manual de usuario y manual técnico.

## FORMA DE ENTREGA

Todo los entregables comprimidos en un archivo .zip con el nombre: [SO2]Practica2\_G<no\_grupo> (Ejemplo: [SO2]Practica2\_G15). Todo debe ir bien identificado con los carnets de los 4 integrantes. Es indispensable que se realice la entrega de 2 formas:

- Mediante UEDI subiendo el archivo zip.

- 
- Subiendo todo a un repositorio de GitHub o GitLab, el cuál debe ser privado con el nombre: SO2\_1S2022\_P2\_G<no\_grupo>, (Ejemplo: SO2\_1S2022\_P2\_G15). Se tendrán que agregar como colaboradores en el repositorio a los usuarios: **Ojr** y **Parobacal**.

Se podrá calificar bien sea utilizando el archivo enviado a UEDI o clonando el repositorio de GitHub, se verificarán las fechas de última actualización de los archivos.

**Fecha de Entrega:** Miércoles 30 de marzo hasta las 23:59.

