

## **Introducción:**

Para el actual Proyecto se debe crear un programa en C diseñado para copiar archivos de un directorio a otro mediante múltiples hilos, implementa un pool de hilos estático que mejora la eficiencia de la copia de archivos, especialmente cuando se trabaja con un gran número de archivos, la copia se hace de manera recursiva y se aplican diferentes pruebas de rendimiento.

## **Descripción del Problema:**

El problema consiste principalmente en copiar todos los archivos de un directorio fuente a un directorio destino de manera eficiente, la copia de archivos puede ser una forma intensiva de gasto de recursos en especial cuando son muchos archivos un único hilo para cada archivo no es eficiente por lo que se implementa un pool

## **Estructuras:**

### **Rutas**

```
typedef struct {  
    char source_path[PATH_MAX];  
    char dest_path[PATH_MAX];  
} file_task;
```

### **Cola de Tareas**

```
typedef struct {  
    file_task tasks[BUFFER_SIZE];  
    int front;  
    int rear;  
    int count;  
    pthread_mutex_t mutex;  
    pthread_cond_t cond;  
} task_queue;
```

## Componentes

```
void init_queue(task_queue *q) {  
    q->front = 0;  
    q->rear = 0;  
    q->count = 0;  
    pthread_mutex_init(&q->mutex, NULL);  
    pthread_cond_init(&q->cond, NULL);  
}
```

## Funciones

### Agrega una tarea a la cola y obtiene una tarea

```
void enqueue(task_queue *q, file_task task) {  
    pthread_mutex_lock(&q->mutex);  
    q->tasks[q->rear] = task;  
    q->rear = (q->rear + 1) % BUFFER_SIZE;  
    q->count++;  
    pthread_cond_signal(&q->cond);  
    pthread_mutex_unlock(&q->mutex);  
}
```

```
file_task dequeue(task_queue *q) {  
    pthread_mutex_lock(&q->mutex);  
    while (q->count == 0) {  
        pthread_cond_wait(&q->cond, &q->mutex);  
    }  
    file_task task = q->tasks[q->front];  
    q->front = (q->front + 1) % BUFFER_SIZE;  
    q->count--;  
    pthread_mutex_unlock(&q->mutex);  
}
```

```
    return task;  
}  
return task;  
}
```

**GitHub:**

<https://github.com/JKevin2713/Sistemas-Operativos.git>