

## **Parcial 2 Sistemas Operativos**

David Alejandro Carreño Parra  
Facultad de ingeniería  
Cuarto semestre – Ing. En sistemas

Universidad Piloto de Colombia  
Bogotá D.C.

19 de abril de 2024

## Implementación:

La implementación la realicé usando una cola de prioridad de prioridad máxima. En este caso, el programa crea un arreglo con 10000 procesos, encola los procesos de este arreglo y luego desencola los procesos y los ejecuta. En mi implementación, cada proceso llama a una función que calcula el factorial de un número que está dentro de los atributos de este proceso, yo hice que todos los procesos calcularan el factorial de 500. Al realizar esto en Python y en C hubieron bastantes diferencias en los tiempos de ejecución, adjunto los resultados:

Para la ejecución en C:

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

726182931448045956649393799686360808350557753012725394593186685462529502318082617070119907410456683967492816929912727369107242537
582500018902039483707674534576812287726807577632021791089881499377434028729548727329453466103125249327552785126258419841562723643
783859909010380865294309721931562803726975876712617975811122234614398103034855769828589539029166167769009017979203291956873688335
272128529069506250476916741687736276700852051766505308818749707345289534690874538248805479112697238597498976390898683562566316274
988932829313201563424917580422645200819707866204431841765171733270055084871252659126603703318074401038386057830776031936512
Proceso 3 en ejecución
Factorial de: 500 = 1220136825991110069739369127662231677560138010034844294127711986324545474132931687149225927106031014375228537
492275621803204464375841730180856867756645377494698549518743114645051614070185672103383615662980615468342802910057499878983059444
309088311713738361017051972754307541444831710773955393053435962880215150932313843566258425124028916935379993365864384360631401865
971022385149579270286593206910810814999411494908002995616911482486765467495685582779997313155452861624565046381505280068670135324
726182931448045956649393799686360808350557753012725394593186685462529502318082617070119907410456683967492816929912727369107242537
582500018902039483707674534576812287726807577632021791089881499377434028729548727329453466103125249327552785126258419841562723643
783859909010380865294309721931562803726975876712617975811122234614398103034855769828589539029166167769009017979203291956873688335
272128529069506250476916741687736276700852051766505308818749707345289534690874538248805479112697238597498976390898683562566316274
988932829313201563424917580422645200819707866204431841765171733270055084871252659126603703318074401038386057830776031936512
Proceso 2 en ejecución
Factorial de: 500 = 1220136825991110069739369127662231677560138010034844294127711986324545474132931687149225927106031014375228537
492275621803204464375841730180856867756645377494698549518743114645051614070185672103383615662980615468342802910057499878983059444
309088311713738361017051972754307541444831710773955393053435962880215150932313843566258425124028916935379993365864384360631401865
971022385149579270286593206910810814999411494908002995616911482486765467495685582779997313155452861624565046381505280068670135324
726182931448045956649393799686360808350557753012725394593186685462529502318082617070119907410456683967492816929912727369107242537
582500018902039483707674534576812287726807577632021791089881499377434028729548727329453466103125249327552785126258419841562723643
783859909010380865294309721931562803726975876712617975811122234614398103034855769828589539029166167769009017979203291956873688335
272128529069506250476916741687736276700852051766505308818749707345289534690874538248805479112697238597498976390898683562566316274
988932829313201563424917580422645200819707866204431841765171733270055084871252659126603703318074401038386057830776031936512
Proceso 1 en ejecución
Factorial de: 500 = 1220136825991110069739369127662231677560138010034844294127711986324545474132931687149225927106031014375228537
492275621803204464375841730180856867756645377494698549518743114645051614070185672103383615662980615468342802910057499878983059444
309088311713738361017051972754307541444831710773955393053435962880215150932313843566258425124028916935379993365864384360631401865
971022385149579270286593206910810814999411494908002995616911482486765467495685582779997313155452861624565046381505280068670135324
726182931448045956649393799686360808350557753012725394593186685462529502318082617070119907410456683967492816929912727369107242537
582500018902039483707674534576812287726807577632021791089881499377434028729548727329453466103125249327552785126258419841562723643
783859909010380865294309721931562803726975876712617975811122234614398103034855769828589539029166167769009017979203291956873688335
272128529069506250476916741687736276700852051766505308818749707345289534690874538248805479112697238597498976390898683562566316274
988932829313201563424917580422645200819707866204431841765171733270055084871252659126603703318074401038386057830776031936512
El programa tardó 1.6784 segundos en ejecutarse
[1] + Done
"/usr/bin/gdb" --interpreter=mi -tty=${DbgTerm} 0<"/tmp/Microsoft-MIEngine-In-g0ffois0.kew" 1>
/tmp/Microsoft-MIEngine-Out-lrvdlojt.com"
david@david-VirtualBox:~/Escritorio/proyectosC$
```

[illegible]

### Análisis:

C es un lenguaje compilado conocido por su rendimiento y eficiencia en la ejecución, mientras que Python es un lenguaje interpretado que prioriza la facilidad de uso y la flexibilidad sobre la velocidad de ejecución pura.

**Compilado vs. Interpretado:** C compila directamente a instrucciones de máquina, lo que significa que su ejecución es directamente sobre el hardware sin capas intermedias significativas. Python, por otro lado, se interpreta en tiempo de ejecución y corre sobre un intérprete (CPython comúnmente), lo que introduce una capa adicional de procesamiento que puede ralentizar la ejecución.

**Gestión de memoria:** C permite un control muy fino sobre la gestión de la memoria, lo que puede contribuir a una ejecución más rápida, especialmente en programas que manejan grandes cantidades de datos o realizan muchas operaciones computacionales. Python automatiza la gestión de la memoria, lo que reduce la complejidad para el programador pero puede aumentar el overhead en tiempo de ejecución.

**Optimización:** Los compiladores de C, como GCC o Clang, realizan una amplia gama de optimizaciones en el código que pueden mejorar significativamente la velocidad de ejecución. Python también realiza algunas optimizaciones, pero estas suelen ser menos efectivas en comparación con las de los compiladores de C.

**Naturaleza de las operaciones:** Las operaciones como calcular un factorial son intensivas computacionalmente y la eficiencia del lenguaje en operaciones matemáticas y de bucles será muy notable. C maneja estas operaciones de manera mucho más eficiente que Python.