Computação Paralela e Distribuída
2023 / 2024

Licenciatura em Engenharia Informática

Trabalho Prático #1 – Números primos grandes

Introdução

Os números primos grandes desempenham um papel essencial em alguns algoritmos de criptografia, onde a segurança depende da dificuldade em factorizar números primos grandes. No entanto, encontrar estes números é um desafio porque não há um padrão ou um método determinístico para os gerar. Pretende-se, com este trabalho, que os alunos explorem várias soluções de forma a encontrar os maiores números primos possíveis.

Desenvolvimento

Como base de partida, o seguinte programa tenta encontrar o maior número primo dentro de um determinado tempo, nomeadamente 5 segundos.

```
import time

def is_prime(n):
    """Check if n is prime."""
    for i in range(2, n-1):
        if n % i == 0:
            return False
    return True

def find_max_prime(timeout):
    """Finds the largest prime until timeout."""
    max_prime = i = 1
    start = time.time()
    while time.time() - start < timeout:
        if is_prime(i) and i > max_prime:
            max_prime = i
        i += 1
    print(max_prime)

if __name__ == '__main__':
    find_max_prime(5)
```

No entanto, os maiores números primos que este programa encontra andam na ordem dos 5 dígitos, o que é manifestamente pouco para os computadores modernos.

Pretende-se melhorar significativamente os resultados. Para tal sugere-se ao alunos melhorar o algoritmo e depois utilizar técnicas de programação paralela de forma a distribuir o trabalho. Os alunos poderão fazer uso de *threads*, processos, variáveis partilhadas e *locks*, consoante a sua interpretação e solução do problema.

A tabela seguinte representa os maiores números primos encontrados numa solução de referência (Macbook Pro CPU 2,4 GHz Intel Core i5 – Dual Core):

Nº Processos	Tempo	Maior primo (nº dígitos)
4	5s	31005333773915239 (17)
4	20s	672578069018335093 (18)
8	20s	4003192130167335 73 (18)
4	60s	3171951615452780311 (19)
8	60s	361590838463309249 (18)

Entrega e avaliação

Os trabalhos deverão ser realizados em grupos de 2 alunos da mesma turma de laboratório, e deverão ser originais. Aos trabalhos plagiados ou cujo código tenha sido partilhado com outros serão atribuídos nota **zero**.

Todos os ficheiros deverão ser colocados num **ficheiro zip** (com os números dos elementos do grupo) e submetidos via moodle **até às 23:55 do dia 05/Maio/2024.** Deverá também ser colocado no *zip* um ficheiro de texto com a identificação dos alunos, alguma descrição da solução que queiram fazer, e uma tabela semelhante à acima com os maiores números primos encontrados para os tempos de 5, 20 e 60 segundos, variando o número de processos/threads (ex: nº cores*1, *2, etc.)

Irá considerar-se a seguinte grelha de avaliação:

Algoritmo base	04 val.
Implementação paralela	04 val.
Utilização correcta de memória partilhada	04 val.
Utilização correcta de mecanismos de sincronização	02 val.
Implementação de melhorias significativas no algoritmo	02 val.
Qualidade da solução e código	04 val.

Bom trabalho!