

**Questão 3 - Threads, processos leves. Escreve dois programas em clang. O primeiro, cria N threads e em seguida, executa *join* para cada um delas. Cada thread deve simplesmente dormir por um tempo (alguns segundos), e em seguida executar *thread\_exit(0)*. Faça um segundo programa, equivalente ao anterior, que cria N processos, que dormem por um tempo determinado, e os esperar terminar. Avalie as diferenças de desempenho, tanto em tempo decorrido para executar as operações importantes), quanto no consumo de memória. Critérios que serão usados na avaliação:**

- **explicação da metodologia experimental**
- **corretude dos resultados obtidos**
- **discussão dos resultados obtidos**

Para analisarmos as diferenças relacionadas a diferença de tempo de criação entre threads e processos, executamos o programa implementado para este fim variando o número de threads e processos criados. Podemos verificar a partir das informações mostradas nas tabelas e gráficos que:

- O tempo gasto na criação de threads é bem menor do que o programa que faz uso de processos.
- Podemos verificar pelos gráficos que em ambos os casos um crescimento linear até a criação de 2000 threads/processos no que se diz respeito ao tempo gasto na criação de processos e threads.
- Como casos extremos podemos observar que o tempo gasto na criação de 10000 threads foi de 44 ms, enquanto para a criação da mesma quantidade de processos o tempo gasto foi de 258 ms.
- No gráfico 3 podemos observar que o tempo da criação dos processos cresce de forma bem mais abrupta que o tempo das threads até o valor de N=2000, onde N é a quantidade de threads/processos criados.

Qtd. Threads	Tempo (em ms)
100	12
150	16
200	18
400	21
600	23
800	28
1000	31
2000	42
3000	44
4000	45
5000	48
6000	47
7000	47
8000	46
9000	49
10000	44

Tabela 1: Número de threads criadas e o tempo gasto (em ms)

Tempo (em ms) - Threads - Executando Join

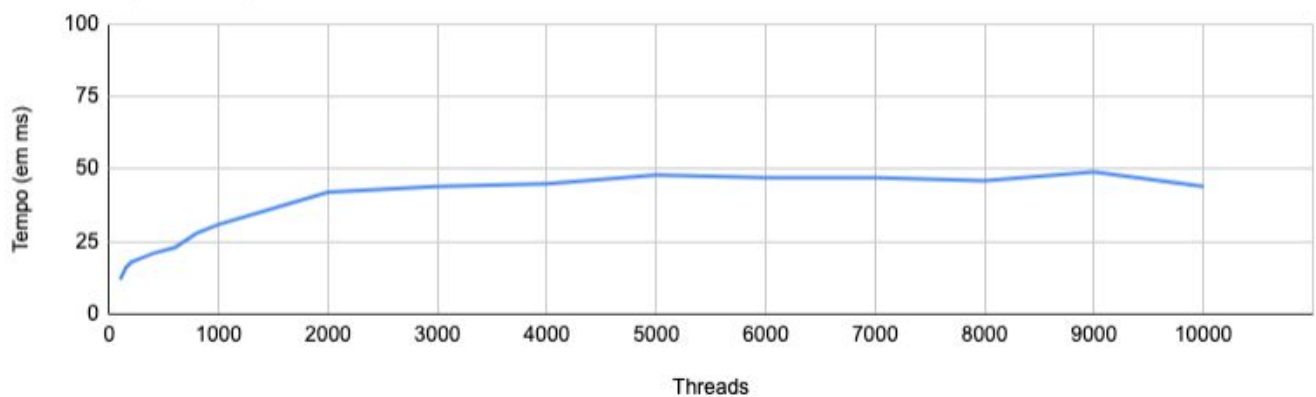


Gráfico 1: Quantidade de threads x Tempo Gasto (ms)

Qtd. Processos	Tempo (em ms)
100	34
150	39
200	54
400	95
600	134
800	182
1000	230
2000	289
3000	265
4000	251
5000	261
6000	259
7000	253
8000	276
9000	261
10000	258

Tabela 2: Número de processos criados e o tempo gasto (em ms)

Tempo (em ms) - Processos - Executando Wait

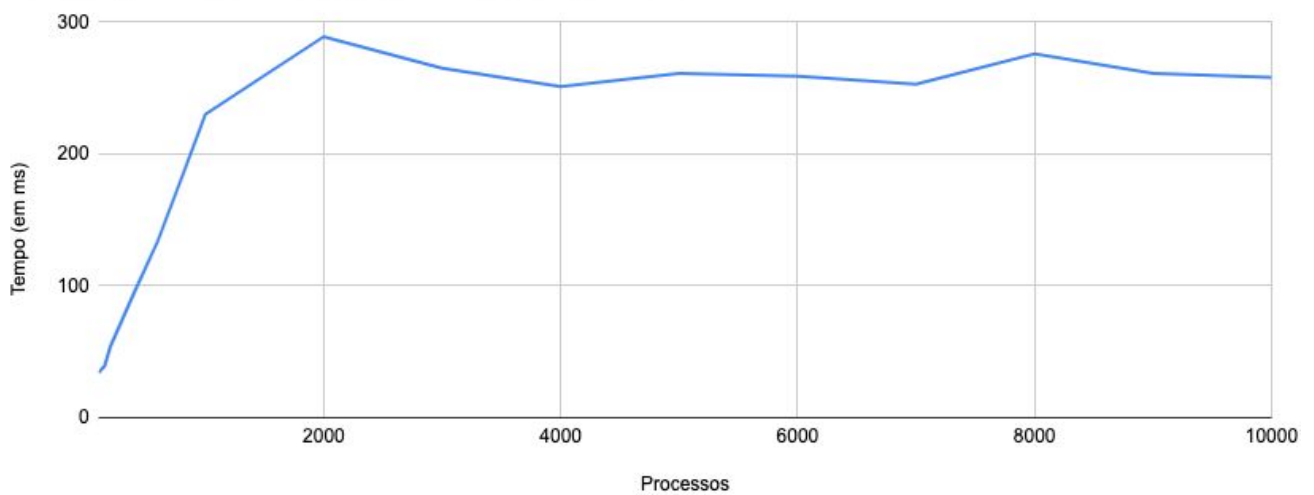


Gráfico 2: Quantidade de processos x Tempo gasto (ms)

Qtd. Processos - Threads	Tempo (em ms) - Processos (Wait)	Tempo (em ms) - Threads (Join)
100	34	12
150	39	16
200	54	18
400	95	21
600	134	23
800	182	28
1000	230	31
2000	289	42
3000	265	44
4000	251	45
5000	261	48
6000	259	47
7000	253	47
8000	276	46
9000	261	49
10000	258	44

Tabela 3: Número de processos e threads criados e o tempo gasto (em ms)

Processos (Wait) e Threads (Join) - Tempo (em ms)

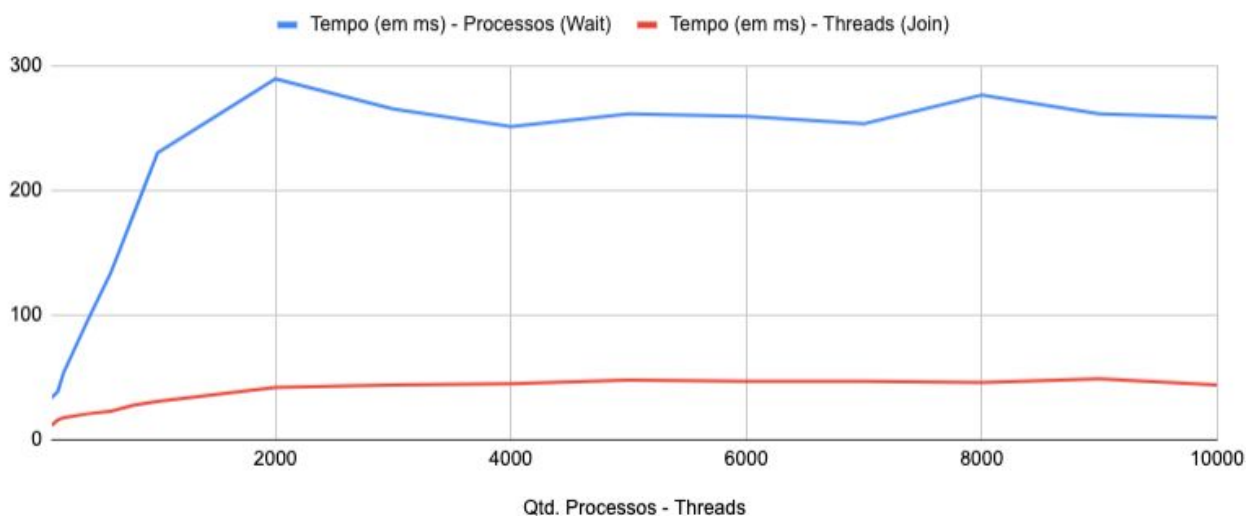


Gráfico 3: Comparação da quantidade de threads processos x Tempo gasto (ms)

Com relação ao consumo de memória observamos que como era de se esperar, a criação de processos consumiu bem mais memória que a criação das threads. Vale salientar que em nosso programa só guardamos o gasto de memória do processo pai, portanto, os dados exibidos referentes ao consumo de memória dos processos é aproximadamente este valor multiplicado pela quantidade de processos criados. Os resultados mostrados abaixo referentes ao consumo de memória estão em bytes.

```

((base) MacBook-Air-de-Neto:q03 gilvanneto$ ./build_start.sh 100
Compilando o código fonte...
- Compilando ex3_p
Executando os binários

Programa criador de 100 Processos
Sou pai! 86841 e esperando filhos de 5 segundos
    5.02 real          0.02 user          0.07 sys
700416 maximum resident set size
    0 average shared memory size
    0 average unshared data size
    0 average unshared stack size
13924 page reclaims
    0 page faults
    0 swaps
    0 block input operations
    0 block output operations
    0 messages sent
    0 messages received
    0 signals received
    72 voluntary context switches
    497 involuntary context switches

Programa criador de 100 Threads
Sou pai! 86944 e esperando filhos de 5 segundos
    5.00 real          0.00 user          0.00 sys
1503232 maximum resident set size
    0 average shared memory size
    0 average unshared data size
    0 average unshared stack size
390 page reclaims
    0 page faults
    0 swaps
    0 block input operations
    0 block output operations
    0 messages sent
    0 messages received
    0 signals received
    0 voluntary context switches
    341 involuntary context switches

```

Figura 1: Consumo de memória para a criação de 100 threads e 1 processo

```

((base) MacBook-Air-de-Neto:q03 gilvanneto$ ./build_start.sh 1000
Compilando o código fonte...
- Compilando ex3_p
Executando os binários

Programa criador de 1000 Processos
Sou pai! 86986 e esperando filhos de 5 segundos
    5.33 real        0.22 user        0.85 sys
 700416 maximum resident set size
      0 average shared memory size
      0 average unshared data size
      0 average unshared stack size
140817 page reclaims
      0 page faults
      0 swaps
      0 block input operations
      0 block output operations
      0 messages sent
      0 messages received
      0 signals received
    495 voluntary context switches
    934 involuntary context switches

Programa criador de 1000 Threads
Sou pai! 87989 e esperando filhos de 5 segundos
    5.02 real        0.01 user        0.08 sys
 8916992 maximum resident set size
      0 average shared memory size
      0 average unshared data size
      0 average unshared stack size
   2200 page reclaims
      0 page faults
      0 swaps
      0 block input operations
      0 block output operations
      0 messages sent
      0 messages received
      0 signals received
      0 voluntary context switches
   3592 involuntary context switches

```

Figura 2: Consumo de memória para a criação de 1000 threads e 1 processo

```

((base) MacBook-Air-de-Neto:q03 gilvanneto$ ./build_start.sh 5000
Compilando o código fonte...
- Compilando ex3_p
Executando os binários

Programa criador de 5000 Processos
Sou pai! 88015 e esperando filhos de 5 segundos
      5.29 real          0.26 user          0.87 sys
704512 maximum resident set size
      0 average shared memory size
      0 average unshared data size
      0 average unshared stack size
158953 page reclaims
      0 page faults
      0 swaps
      0 block input operations
      0 block output operations
      0 messages sent
      0 messages received
      0 signals received
    329 voluntary context switches
    8870 involuntary context switches

Programa criador de 5000 Threads
Sou pai! 89135 e esperando filhos de 5 segundos
      5.03 real          0.02 user          0.16 sys
17563648 maximum resident set size
      0 average shared memory size
      0 average unshared data size
      0 average unshared stack size
   7263 page reclaims
      0 page faults
      0 swaps
      0 block input operations
      0 block output operations
      0 messages sent
      0 messages received
      0 signals received
      0 voluntary context switches
   6882 involuntary context switches

```

Figura 3: Consumo de memória para a criação de 5000 threads e 1 processo

```

((base) MacBook-Air-de-Neto:q03 gilvanneto$ ./build_start.sh 10000
Compilando o código fonte...
- Compilando ex3_p
Executando os binários

Programa criador de 10000 Processos
Sou pai! 92641 e esperando filhos de 5 segundos
      5.30 real          0.29 user          0.89 sys
724992 maximum resident set size
      0 average shared memory size
      0 average unshared data size
      0 average unshared stack size
158781 page reclaims
      0 page faults
      0 swaps
      0 block input operations
      0 block output operations
      0 messages sent
      0 messages received
      0 signals received
    441 voluntary context switches
    8820 involuntary context switches

Programa criador de 10000 Threads
Sou pai! 93759 e esperando filhos de 5 segundos
      5.04 real          0.04 user          0.20 sys
17653760 maximum resident set size
      0 average shared memory size
      0 average unshared data size
      0 average unshared stack size
  12296 page reclaims
      0 page faults
      0 swaps
      0 block input operations
      0 block output operations
      0 messages sent
      0 messages received
      0 signals received
      0 voluntary context switches
    7168 involuntary context switches

```

Figura 4: Consumo de memória para a criação de 10000 threads e 1 processo