

Atividade 1

Rodei até 15 milhões, pareceu tudo certo. Conferi visualmente com https://www.walter-fendt.de/html5/mpt/primenumbers_pt.htm

Atividade 3

No lugar do teste de primalidade simplão, implementei o teste de Miller-Rabin. Ele é um teste probabilístico de primalidade. No programa, utilizamos 10 iterações, que, pelo teorema de Rabin, garante a probabilidade de erro de $1 - (1/4)^{10} > 1e-6$.

A outra modificação foi que implementei a lista de futuros de forma circular. Dessa forma, mesmo com N grande, é possível executar o programa. Ele não coloca as N tarefas no pull de threads imediatamente. Coloca apenas `10*NTHREADS` tarefas na lista e remove uma vez que foram concluídas.

Com a modificação da lista, a aplicação ficou estável em ~150mb de consumo de memória.

Primos de 1 a 1bi

Não adicionarei ao repositório porque o arquivo ficou com mais de 500MB. Para se ter uma ideia da dimensão, o comando `cat` levou mais de 35s para imprimir todos eles.

Nestes termos, a execução do programa levou ~5min.

Observe que há bastante espaço para melhoria, a começar por não verificar números pares...

Todos

- ☒ Atividade 1, item 4
- ☒ Atividade 3, item 2
- ☒ Atividade 3, item 3
- ☒ Implementar a lista de futuros com tamanho fixo, menor que N para evitar erros de alocação de memória.