

readme.md

Tarefa 2

Aluno: Rafael Avraham Gildin Acherboim - 31940242

Exercícios com processos:

- i. Rode o programa anterior para valores grandes de n. As mensagens sempre estarão ordenadas pelo valor de i?
- ii. O que acontece se o programa anterior escreve-se as mensagens para sys.stdout, usando print, ao invés de para sys.stderr?

Respostas:

1. Sim. Pois com o comando sys.stderr as mensagens são enviadas como se fossem erros, logo são enviadas imediatamente quando ocorrem, o que mantém a ordem dessas.

```
● rafagildin@Rafael-Gildin:~/git/computacao-paralela/aula2/codigo-aula$ gcc cadeia_processos.c -o cadeia_processos && ./cadeia_processos 20
i:1 process ID:4443 parent ID:728 child ID:4444
i:2 process ID:4444 parent ID:160 child ID:4445
i:3 process ID:4445 parent ID:4444 child ID:4446
i:4 process ID:4446 parent ID:4445 child ID:4447
i:5 process ID:4447 parent ID:4446 child ID:4448
i:6 process ID:4448 parent ID:4447 child ID:4449
● rafagildin@Rafael-Gildin:~/git/computacao-paralela/aula2/codigo-aula$ i:7 process ID:4449 parent ID:4448 child ID:4450
i:8 process ID:4450 parent ID:4449 child ID:4451
i:9 process ID:4451 parent ID:4450 child ID:4452
i:10 process ID:4452 parent ID:4451 child ID:4453
i:11 process ID:4453 parent ID:4452 child ID:4454
i:12 process ID:4454 parent ID:4453 child ID:4455
i:13 process ID:4455 parent ID:4454 child ID:4456
i:14 process ID:4456 parent ID:4455 child ID:4457
i:15 process ID:4457 parent ID:4456 child ID:4458
i:16 process ID:4458 parent ID:4457 child ID:4459
i:17 process ID:4459 parent ID:4458 child ID:4460
i:18 process ID:4460 parent ID:4459 child ID:4461
i:19 process ID:4461 parent ID:4460 child ID:4462
i:20 process ID:4462 parent ID:4461 child ID:0
● rafagildin@Rafael-Gildin:~/git/computacao-paralela/aula2/codigo-aula$
```

2. As mensagens estão desordenadas, uma vez que com o comando sys.stdout elas são armazenadas em um buffer antes de serem printadas na tela, o que pode gerar a desordenação destas.

```
● rafagildin@Rafael-Gildin:~/git/computacao-paralela/aula2$ gcc cadeia_processos.c -o cadeia_processos && ./cadeia_processos 20
i:1 process ID:4718 parent ID:728 child ID:4719
i:2 process ID:4719 parent ID:4718 child ID:4720
i:3 process ID:4720 parent ID:4719 child ID:4721
i:4 process ID:4721 parent ID:4720 child ID:4722
i:5 process ID:4722 parent ID:4721 child ID:4723
i:6 process ID:4723 parent ID:4722 child ID:4724
● rafagildin@Rafael-Gildin:~/git/computacao-paralela/aula2$ i:7 process ID:4724 parent ID:160 child ID:4725
i:8 process ID:4725 parent ID:4724 child ID:4726
i:9 process ID:4726 parent ID:4725 child ID:4727
i:10 process ID:4727 parent ID:160 child ID:4728
i:11 process ID:4728 parent ID:4727 child ID:4729
i:12 process ID:4729 parent ID:4728 child ID:4730
i:13 process ID:4730 parent ID:160 child ID:4731
i:14 process ID:4731 parent ID:4730 child ID:4732
i:15 process ID:4732 parent ID:4731 child ID:4733
i:16 process ID:4733 parent ID:4732 child ID:4734
i:17 process ID:4734 parent ID:4733 child ID:4735
i:18 process ID:4735 parent ID:160 child ID:4736
i:20 process ID:4737 parent ID:4736 child ID:0
i:19 process ID:4736 parent ID:4735 child ID:4737
```