

# Forum:[Hack Forums](#)

Thread: [Closing file pointers increases run time.](#)

**SIM-888979** - November 20, 2020, 8:29 am

Hi guys, I'm currently writing a program that will run some calculations then store the data into a .txt file. I do not know how many iterations or data I would have so I cannot use an array to store the data first then process later. So, I decided to call the write to file function every time I do a calculation. This allows me to write up to about 507 lines of data before the buffer becomes full and then refusing to write any more lines. I can have 1 line or I can have 10000 lines in my data, so it is very important I capture all the data.

I know I can fix this by just closing the file using fclose but I also need to keep to a low run time. By not doing fclose(), I cut my runtime from 8 seconds to 0.4 seconds for about 900 lines. What should I do, how do I go about keeping a low run time but still capturing all data in a .txt file?

Code void gnuplot(float mae, int iteration) {

```
FILE *plot_ptr;
plot_ptr = fopen("plotting-data.txt", "a+");

fprintf(plot_ptr, "%i %f\n", iteration, mae);
printf("Write complete.");
}
```

Fixed! I realized I think each time I called the function, I opened a new file pointer each time? Not too sure. I just decided to use a single file pointer in the main function and parse the pointer to it instead.

**Donkey Death Jam** - November 20, 2020, 10:03 am

(Nov 20, 2020 09:29 AM) Lotus Wrote: [\[CITE\]](#) Fixed! I realized I think each time I called the function, I opened a new file pointer each time? Not too sure. I just decided to use a single file pointer in the main function and parse the pointer to it instead.

If gnuplot() is called within the loop, then a new file handle would have indeed been created for every iteration. Glad you've managed to fix it!

**SIM-888979** - November 20, 2020, 10:23 am

(Nov 20, 2020 11:03 AM) Donkey Death Jam Wrote: [\[CITE\]](#) If gnuplot() is called within the loop, then a new file handle would have indeed been created for every iteration. Glad you've managed to fix it! Yes, I never realized that. Thank you for confirming. ?

**G.P.D.** - November 21, 2020, 8:14 am

You could also split your calculations into chunks.

Let's say you have 10k calculations to do, then write it 1k per 1k onto your file.

It should improve the performance as well if I'm not wrong, it's been a long time since I wrote a line a line of C.

**0xDEAD10CC** - November 22, 2020, 10:58 am

Why wouldn't you pass the FILE pointer into the function if you're calling it from a loop?

**SIM-888979** - November 22, 2020, 5:29 pm

(Nov 22, 2020 11:58 AM)0xDEAD10CC Wrote: [\[CITE\]](#) Why wouldn't you pass the FILE pointer into the function if you're calling it from a loop?

First time working with pointers. Though I fixed this already by just passing the file pointer as you mentioned. ?