OS 2019 Problem Sheet 1 Course: CO20-320202

Arsenij Percov

September 23, 2019

1. a) Check file scat.c.

   b)To answer this question I generated file of 40kb of plain text.
   I wrote a small script to run each loop 10 times for that file and measure its time.
   This is result for library calls:

   ```
   real 0m0.007s user 0m0.007s sys 0m0.000s
   real 0m0.003s user 0m0.003s sys 0m0.000s
   real 0m0.003s user 0m0.003s sys 0m0.000s
   real 0m0.003s user 0m0.003s sys 0m0.000s
   real 0m0.003s user 0m0.003s sys 0m0.000s
   real 0m0.003s user 0m0.003s sys 0m0.000s
   real 0m0.003s user 0m0.003s sys 0m0.000s
   real 0m0.003s user 0m0.003s sys 0m0.000s
   real 0m0.003s user 0m0.002s sys 0m0.000s
   real 0m0.003s user 0m0.002s sys 0m0.000s
   real 0m0.002s user 0m0.002s sys 0m0.000s
   ```

   This is result for system calls:

   ```
   real 0m0.390s user 0m0.143s sys 0m0.247s
   real 0m0.391s user 0m0.153s sys 0m0.235s
   real 0m0.390s user 0m0.155s sys 0m0.231s
   real 0m0.390s user 0m0.146s sys 0m0.241s
   real 0m0.392s user 0m0.180s sys 0m0.209s
   real 0m0.410s user 0m0.165s sys 0m0.245s
   real 0m0.424s user 0m0.256s sys 0m0.168s
   real 0m0.409s user 0m0.188s sys 0m0.221s
   real 0m0.402s user 0m0.151s sys 0m0.251s
   real 0m0.395s user 0m0.156s sys 0m0.240s
   real 0m0.394s user 0m0.152s sys 0m0.240s
   ```

   We can obviously see the difference between the execution time, and conclude that library call is drastically faster. Especially, for system calls loop we can notice that more than half of the time was taken by system calls. I repeated the results multiple time and results were the same. After that I traced the system calls. And it gave us good view on the situation. The library call is optimized in the background and we write 4096 bytes per system call. That requires less system calls, and spends less time on them and transitions of modes. The system call loop however was calling the individual system call for every byte. That resulted in 4096 times more calls than compared to the library call loop. That takes drastically more time, and we observed that result by measuring the times. For library calls 98 read system calls and 98 write system calls were called. For system calls loop, we executed 400000 of each system call (read and write). You can find strace result attached in the folder (straceres.txt)

c) We chose 4096 bytes as we mentioned in b. We measured the execution times for the same file with the same script:

```
real 0m0.002s user 0m0.002s sys 0m0.000s
real 0m0.002s user 0m0.002s sys 0m0.000s
real 0m0.002s user 0m0.000s sys 0m0.002s
real 0m0.002s user 0m0.001s sys 0m0.000s
real 0m0.002s user 0m0.000s sys 0m0.002s
real 0m0.002s user 0m0.001s sys 0m0.000s
real 0m0.002s user 0m0.000s sys 0m0.002s
real 0m0.002s user 0m0.001s sys 0m0.000s
real 0m0.002s user 0m0.001s sys 0m0.000s
real 0m0.001s user 0m0.001s sys 0m0.000s
real 0m0.001s user 0m0.001s sys 0m0.000s
```

This is the fastest way to send data known. This is the only copying that is done in kernel, so it is more efficient that read() and write() system calls. The reason for that is eliminating the intermediate buffer and saving time spent on transferring data there.(Linux manual)

2. Check file watch.c