Universidade Federal de Uberlândia

2° Atividade Bônus

Trabalho redigido na disciplina sistemas operacionas,

dirigido pelo professor Rivalino Matias Júnior.

João Paulo de Oliveira 11611BCC046

Uberlândia

2017

Objetivo do experimento

Comparar a velocidade de um programa que imprime uma réplica da entrada usando system call, e outro programa que realiza a mesma operação utilizando a biblioteca "*stdio.h*" da linguagem.

Ambiente

- Versão do Kernel: 4.9.0-3
- Distribuição: Debian GNU/Linux 9 (stretch) 64-bit
- Processador: Intel® Core™ i5-3337U CPU @ 1.80GHz × 4
- Memória Ram: 5.7 GiB (6.0 Gb)

Procedimentos adotados

Foi criado 2 programas:

- libcx:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
void main(int argc, char* argv[]){
   int x = (int) pow(2,atoi(argv[1]));
   char *buffer = (char*)malloc(sizeof(char)* x);
   int i;
   i = fread(buffer, x, 1, stdin);
   while (i>0){
      fwrite(buffer, x, 1, stdout);
      i = fread(buffer, x, 1, stdin);
   }
}
```

- syscallx:

```
#include <unistd.h>
#include <stdlib.h>
#include <math.h>
main(int argc, char* argv[]){
   int x = pow(2,atoi(argv[1]));
   char *buffer = (char*)malloc(sizeof(char)* x);
   int i;
   i = read(0,buffer,x);
   while (i>0){
      write(1,buffer,x);
      i = read(0,buffer,x);
   }
}
```

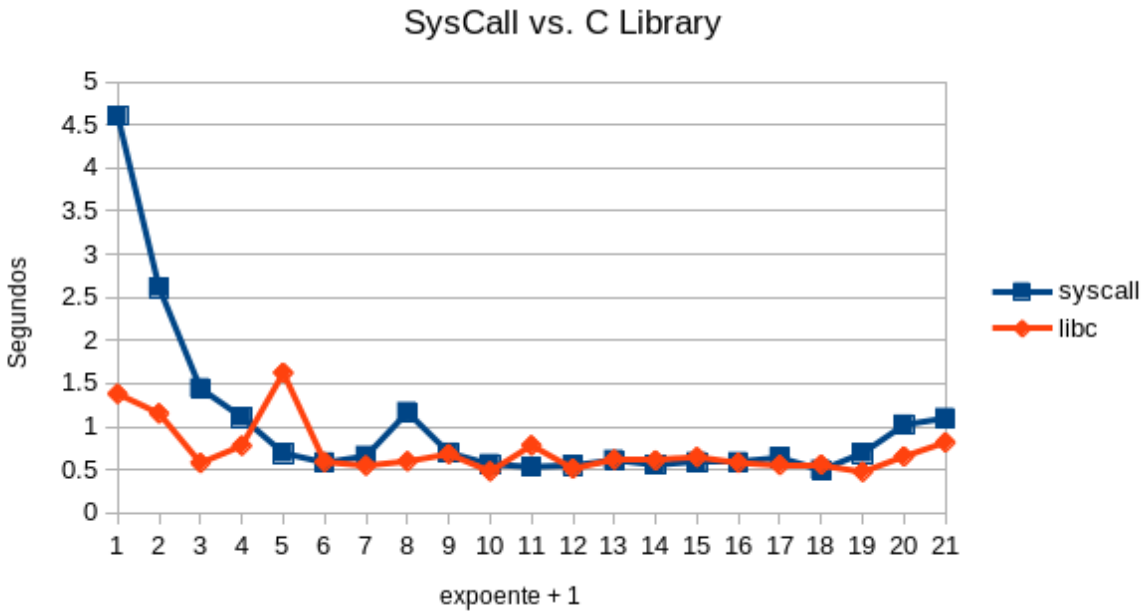
Esses programas foram criados de forma a serem quase idênticos, mudando apenas a biblioteca e a função chamada para ler e escrever os dados. No terminal, foi utilizado as seguintes linhas de comando:

- *sync; echo 3 > /proc/sys/vm/drop_caches; time ./libcx X < datafile > /dev/null*
- *sync; echo 3 > /proc/sys/vm/drop_caches; time ./syscallx X < datafile > /dev/null*

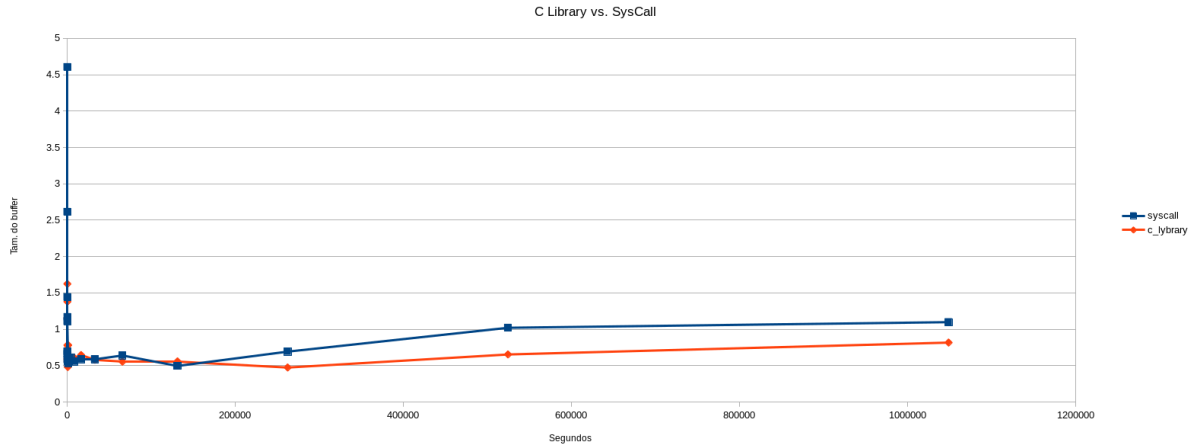
Nos quais X é o tamanho da variável buffer e o arquivo datafile foi criado com o comando: "*dd if=/dev/zero of=/home/datafile bs=1024 count=10K*" que resultou um arquivo com 10.5 MB.

Resultados

Os dados foram cruzados resultando no seguinte gráfico:



Onde o eixo X é o expoente de 2 somado com 1, para facilitar a visualização. Mas o gráfico original é:



Os resultados quer produziram o gráfico foram:

	A	B	C
1	1	4.604	1.379
2	2	2.614	1.153
3	4	1.442	0.578
4	8	1.109	0.779
5	16	0.692	1.624
6	32	0.583	0.59
7	64	0.658	0.548
8	128	1.168	0.598
9	256	0.696	0.681
10	512	0.564	0.481
11	1014	0.534	0.782
12	2048	0.551	0.513
13	4096	0.612	0.618
14	8192	0.556	0.607
15	16384	0.587	0.646
16	32768	0.586	0.58
17	65536	0.64	0.555
18	131072	0.496	0.558
19	262144	0.693	0.474
20	524288	1.021	0.654
21	1048576	1.098	0.817
22	buffer	SysCall	C Library

Conclusões

Há ganho de desempenho quando se opta pelas bibliotecas da linguagem C, pois as system calls não acessam diretamente as funções do kernel, elas geralmente passam por uma API. A biblioteca "*unistd.h*", usada no programa, tem o padrão POSIX, e aí sim são chamadas as funções. O caso das bibliotecas do C são mais rápidas, pois a função ocorre em tempo de usuário e passa mais rapidamente para o kernel. Elas são feitas assim justamente para ganhar tempo e não precisam chamar uma API intermediária para resolver algumas indefinições, como por exemplo, "chamarei o '*write()*' do NTFS ou do Ext2?".