

Laboratório 20

Chamadas ao Sistema Operacional

1. Baixe o programa “copia.c” do Moodle. Esse programa copia um arquivo informado em linha de comando para a saída padrão (tela). Compile e execute o programa, testando-o com alguns arquivos de entrada.

2. Observe o código de “copia.c”. A função *main()* chama quatro funções definidas no mesmo arquivo: *myopen()*, *myread()*, *mywrite()* e *myclose()*. Cada uma dessas funções é implementada por uma única linha, que contém uma chamada ao sistema operacional (na verdade, uma chamada a uma função wrapper, provida pela biblioteca padrão de C).

Caso tenha dúvida nas chamadas ao sistema operacional, use a documentação na seção 2 do manual do Linux. Para ver a documentação de uma delas, basta escrever no terminal de comandos:

```
$ man 2 open
```

Para sair do manual, digite “q” (de *quit*).

3. Substitua cada uma das funções (*myopen*, *myread*, *mywrite* e *myclose*) por funções equivalentes, mas escritas em Assembly. Use a seguinte linha para compilar o novo programa:

```
gcc -Wall -o copia chamadas.s copia.c
```

As funções Assembly não devem chamar as funções da biblioteca de C. Elas devem usar uma interface direta com o SO. Crie um arquivo “chamadas.s” e comece substituindo uma função em C por vez, por exemplo, comece com a função *myopen*: comente a implementação dessa função no arquivo “copia.c”, mantendo o seu protótipo, e faça sua implementação em Assembly. Faça isso para as demais funções.

Para saber o código das chamadas ao sistema, consulte o [link](#) para a tabela apresentada nos slides da aula, bem como o procedimento que deve ser realizado.

Ao final, a implementação em Assembly de chamadas ao sistema operacional que você implementar será semelhante à implementação das funções wrapper disponíveis na biblioteca padrão de C.

4. Traduza somente a função *mysleep()* abaixo para Assembly, mas em vez de chamar a função *nanosleep()*, faça uma chamada de sistema *sys_nanosleep* diretamente ao SO.

Use o comando “*man nanosleep*” para ver os campos da estrutura *timespec*. Considere o tipo “*time_t*” como sendo um *long* de C.

Para testar, comente a implementação de *mysleep()* (lembre-se de deixar o protótipo da função). Por causa da função *modf()*, é necessário compilar o programa com a biblioteca matemática “-lm”.

```
gcc -o mysleep main.c mysleep.s -lm
```

```
#include <stdio.h>
#include <time.h>
#include <math.h>

int mysleep(double t);

int mysleep(double t)
{
    double ipart;
    double fpart;
    struct timespec req;
    struct timespec rem;

    fpart = modf(t, &ipart);

    req.tv_sec = (time_t)ipart;
    req.tv_nsec = (time_t)(fpart * 1.0E9); // seg -> nano

    return nanosleep(&req, &rem);
}

int main() {
    // Dormir 7 segundos e 465 milisegundos
    mysleep(7.465);
    return 0;
}
```

