Nome arquivo: my\_semaphore.c

sem\_wait - [link](https://man7.org/linux/man-pages/man3/sem_wait.3.html)

**sem\_wait** () decrementa (bloqueia) o semáforo apontado por *sem* . Se

o valor do semáforo for maior que zero, o decremento

continua e a função retorna imediatamente. Se o semáforo

atualmente tem o valor zero, então a chamada bloqueia até que

seja possível realizar o decremento (ou seja, o valor do semáforo

sobe acima de zero) ou um manipulador de sinal interrompe a chamada.

Nome arquivo: aplicacao.c

sem\_init - [link](https://www.man7.org/linux/man-pages/man3/sem_init.3.html)

Inicializa um semáforo sem nome. Inicializa o semáforo sem nome no endereço apontado para por *sem* . O argumento do *valor* especifica o valor inicial para o semáforo. **sem\_init** () retorna 0 em caso de sucesso; em caso de erro, -1 é retornado, e [*errno*](https://www.man7.org/linux/man-pages/man3/errno.3.html)

é definido para indicar o erro.

sem\_post - [link](https://man7.org/linux/man-pages/man3/sem_post.3.html)

Função - Desbloquear um semáforo.

**sem\_post** () incrementa (desbloqueia) o semáforo apontado por *sem* . E se

o valor do semáforo consequentemente torna-se maior do que zero, então

outro processo ou thread bloqueado em uma [chamada sem\_wait (3)](https://man7.org/linux/man-pages/man3/sem_wait.3.html) será ativado

e prossiga para bloquear o semáforo.

## **VALOR DE RETORNO**

**sem\_post** () retorna 0 em caso de sucesso; em caso de erro, o valor do semáforo

é deixado inalterado, -1 é retornado e [*errno*](https://man7.org/linux/man-pages/man3/errno.3.html) é definido para indicar o

erro.

**dlsym ()** [**link**](https://linux.die.net/man/3/dlsym)

A função **dlsym** () pega um "manipulador" de uma biblioteca dinâmica retornada por **dlopen** () e o nome do símbolo terminado em nulo, retornando o endereço onde aquele símbolo é carregado na memória. Se o símbolo não for encontrado, na biblioteca especificada ou em qualquer uma das bibliotecas que foram carregadas automaticamente por **dlopen** () quando essa biblioteca foi carregada, **dlsym** () retorna NULL. (A pesquisa realizada por **dlsym** () é ampla primeiro através da árvore de dependências dessas bibliotecas.) Como o valor do símbolo pode ser NULL (de forma que um retorno NULL de **dlsym** () não precisa indicar um erro), a maneira correta testar se há um erro é chamar **dlerror**() para limpar quaisquer condições de erro antigas, chame **dlsym** () e depois chame **dlerror** () novamente, salvando seu valor de retorno em uma variável e verifique se esse valor salvo não é NULL.

pthread\_exit() - [link](https://www.man7.org/linux/man-pages/man3/pthread_exit.3.html)

Termina o thread de chamada.

pthread\_create - [link](https://man7.org/linux/man-pages/man3/pthread_create.3.html)

Cria um novo tópico. A função **pthread\_create** () inicia uma nova thread na chamada

processo. O novo thread inicia a execução invocando

*start\_routine* (); *arg* é passado como o único argumento de

*start\_routine* ().

pthread\_join - [link](https://man7.org/linux/man-pages/man3/pthread_join.3.html)

Junta-se com um tópico encerrado.

A função **pthread\_join** () espera pelo thread especificado por *thread*

terminar. Se esse tópico já terminou, então

**pthread\_join** () retorna imediatamente. O tópico especificado por *tópico*

deve ser juntável.

Se *retval* não for NULL, então **pthread\_join** () copia o status de saída de

o segmento de destino (ou seja, o valor que o segmento de destino forneceu para

[pthread\_exit (3)](https://man7.org/linux/man-pages/man3/pthread_exit.3.html) ) no local apontado por *retval* . Se o

thread de destino foi cancelado, então **PTHREAD\_CANCELED** é colocado no

localização apontada por *retval* .

Se vários tópicos simultaneamente tentarem se juntar ao mesmo tópico,

os resultados são indefinidos. Se o thread que chama **pthread\_join** () é

cancelado, então o segmento de destino permanecerá juntável (ou seja,

não ser desanexado).

## **VALOR DE RETORNO** [**topo**](https://man7.org/linux/man-pages/man3/pthread_join.3.html#top_of_page)

Em caso de sucesso, **pthread\_join** () retorna 0; em caso de erro, ele retorna um erro

número.

pthread\_self () retorna o ID da thread na qual é chamada