

SEMÁFOROS BINÁRIOS NO LINUX USANDO C

- . **Mutex**, implementado na biblioteca **pthread**
- . Uso mais simples que semáforo
- . Restrito ao próprio processo (e *threads* correspondentes)
- . Uso:
 - . Defina variável do tipo **pthread_mutex_t** **mutex**
 - . Use a função
pthread_mutex_lock(&mutex)
para executar um **wait** no mutex

- . Use a função

pthread_mutex_unlock(&mutex);

para executar um signal no mutex

Exemplo:

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

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

```
#include <pthread.h>
```

```
pthread_t t1,t2; // duas threads
```

```
pthread_mutex_t sb; //semaforo mutex => garante exclusao mutua
```

```
int x=1000; //variavel global compartilhada
```

```
void *thread1(void *par){  
    int y;  
    pthread_mutex_lock(&sb); // entra na regioao critica  
    y = x;  
    printf("Thread 1 leu x=%d\n",y);  fflush(stdout);  
    sleep(1);  
    y = y + 50;  
    printf("Thread 1 gravou x=%d\n",y); fflush(stdout);  
    x = y;  
    pthread_mutex_unlock(&sb); // sai da regioao critica  
}
```

```
void *thread2(void *par){  
    int z;  
    pthread_mutex_lock(&sb); // entra na regioao critica  
    z = x;  
    sleep(1);  
    printf("Thread 2 leu x=%d\n",z); fflush(stdout);  
    z = z + 30;  
    printf("Thread 2 gravou x=%d\n",z); fflush(stdout);  
    x = z;  
    pthread_mutex_unlock(&sb); // sai da regioao critica  
}
```

```
int main(void){  
    pthread_create(&t1,NULL,&thread1,NULL);  
    pthread_create(&t2,NULL,&thread2,NULL);  
    pthread_join(t1,NULL);  
    pthread_join(t2,NULL);  
    printf("Acabou com x=%d\n",x); fflush(stdout);  
    return 0;  
}
```