

MC514–Sistemas Operacionais: Teoria e Prática
1s2010

Barreiras

O ponto de encontro das threads

- Grupo de N threads
- Vida monótona
 - Trabalham
 - Sincronização na barreira
- Vários exemplos retirados do livro The Little Book of Semaphores
- Como ser mais eficiente?

Primeira tentativa (3.4)

```
int c;  
sem_barreira = 0;  
atomic_inc(c);  
if (c == N) sem_post(barreira);  
sem_wait(barreira);
```

- Quais são os problemas deste código?
- Quando funciona?
- Veja o arquivo `barreira1.c`

Segunda tentativa (3.5)

```
int c;  
sem_barreira = 0;  
atomic_inc(c);  
if (c == N) sem_post(barreira);  
sem_wait(barreira);  
sem_post(barreira);
```

- Veja o arquivo `barreira2.c`
- Se você acha que está ok veja `barreira2b.c`

Terceira tentativa (3.7)

```
int c;  
sem_barreira = 0;  
atomic_inc(c);  
if (c == N) sem_post(barreira);  
sem_wait(barreira);  
sem_post(barreira);  
atomic_dec(c);  
if (c == 0) sem_wait(barreira);
```

- Veja o arquivo barreira3.c

Quarta tentativa (3.8)

```
int c;  
sem_barreira = 0;  
local_c = atomic_inc(c);  
if (local_c == N) sem_post(barreira);  
sem_wait(barreira);  
sem_post(barreira);  
local_c = atomic_dec(c);  
if (local_c == 0) sem_wait(barreira);
```

- Veja o arquivo barreira4.c

Quinta tentativa (3.8)

Roleta dupla

```
int c;  
sem_roleta_entrada = 0, roleta_saida = 1;  
local_c = atomic_inc(c);  
if (local_c == N)  
    sem_wait(roleta_saida);  
    sem_post(roleta_entrada);  
  
sem_wait(roleta_entrada);  
sem_post(roleta_entrada);
```

Quinta tentativa (3.8)

Roleta dupla (continuação)

```
local_c = atomic_dec(c);  
if (local_c == 0)  
    sem_wait(roleta_entrada);  
    sem_post(roleta_saida);
```

```
sem_wait(roleta_saida);  
sem_post(roleta_saida);
```

- Veja o arquivo barreira5.c

Deadlock (3.6)

```
int c; lock_t mutex;  
sem barreira = 0;  
lock(mutex);  
c++;  
if (c == N) sem_post(barreira);  
sem_wait(barreira);  
sem_post(barreira);  
unlock(mutex);
```

- Uso claro para variáveis de condição...

Sobre desempenho...

- Duas roletas causam muitas trocas de contexto
- Na seção 3.6.6 o livro sugere um semáforo de incremento maior do que 1
- Variáveis de condição e broadcast?
- Futex?
- E a implementação da libc? Como é?

Tentativa com futex

Roleta dupla

```
int c;  
int roleta_entrada = 0, roleta_saida = 1;  
local_c = atomic_inc(c);  
if (local_c == N)  
    roleta_saida = 0;  
    roleta_entrada = 1;  
    futex_wake(&roleta_entrada, N-1);  
else  
    futex_wait(&roleta_entrada, 0);
```

Tentativa com futex

Roleta dupla (continuação)

```
local_c = atomic_dec(c);  
if (local_c == 0)  
    roleta_entrada = 0;  
    roleta_saida = 1;  
    futex_wake(&roleta_saida, N-1);  
else  
    futex_wait(&roleta_saida, 0);
```

- Veja o arquivo `barreira_futex.c`

Pthread_barrier_wait

- Qual algoritmo está implementado?
- Veja o código `pthread_barrier_wait` e o teste `barrier_wait.c`