

Sistemas Operacionais

Leitores e Escritores

João Lucas e Luis Henrique

Função Reader

```
8   sem_t mutex;           /* controla o acesso a 'rc' */
9   sem_t db;              /* controla o acesso a base de dados */
10  int rc = 0;             /* número de processos lendo ou querendo ler */
11  int dbv;                /* valor na base dados */
12
13  void* reader(void *arg) {
14      int i = *((int *) arg);
15
16      while(1) {           /* repete para sempre */
17          sem_wait(&mutex); /* obtém acesso exclusivo à 'rc' */
18          rc = rc + 1;      /* um leitor a mais agora */
19
20          if (rc == 1) {    /* se este for o primeiro leitor... */
21              sem_wait(&db);
22          }
23
24          sem_post(&mutex); /* libera o acesso exclusivo a 'rc' */
25          read_data_base(i); /* acesso aos dados */
26          sem_wait(&mutex); /* obtém acesso exclusivo a 'rc' */
27          rc = rc - 1;      /* um leitor a menos agora */
28
29          if (rc == 0) {    /* se este for o último leitor */
30              sem_post(&db);
31          }
32
33          sem_post(&mutex); /* libera o acesso exclusivo a 'rc' */
34          use_data_read(i); /* região não crítica */
35      }
36  }
```

Função Writer

```
38 void* writer(void *arg) {  
39     int i = *((int *) arg);  
40     int value;  
41  
42     while(1) {  
43         value = think_up_data(i);  
44         sem_wait(&db);  
45         write_data_base(i, value);  
46         sem_post(&db);  
47     }  
48 }  
49
```

read_data, use_data, think_up

```
50 void read_data_base(int i) {
51     srand(time(0));
52     printf("Reader %d leu o valor %d na base de dados!\n", i+1, dbv);
53     delay( rand() % 5);
54 }
55
56 void use_data_read(int i) {
57     srand(time(0));
58     printf("Reader %d estah usando os dados lidos!\n", i+1);
59     delay(rand() % 5);
60 }
61
62 int think_up_data(int i) {
63     srand(time(0));
64     printf("Writer %d estah pensando no que escrever!\n", i+1);
65     delay(rand() % 5);
66     return rand()%100;
67 }
```

write_data_base

```
69 void write_data_base(int i, int value) {  
70     srand(time(0));  
71     dbv = value;  
72     printf("Writer %d escreveu o valor %d nos dados!\n", i+1, value);  
73     delay( rand() % 5);  
74 }
```

Main

```
89 int main() {
90     sem_init(&mutex, 0, 1);
91     sem_init(&db, 0, 1);
92
93     pthread_t r[3], w[2];
94
95     int i;
96     srand(time(0));
97     dbv=rand()%100; /*inicia dbv*/
98
99     /* criando leitores */
100     for (i = 0; i < 3 ; i++) {
101         pthread_create(&r[i], NULL, reader, (void*) &i);
102         delay(1);
103     }
104
105     /* criando escritores */
106     for (i = 0; i < 2; i++) {
107         pthread_create(&w[i], NULL, writer, (void*) &i);
108         delay(1);
109     }
110
111     while(1);
112
113     return 0;
114 }
115
```