
MAC0438 – Programação concorrente

Daniel Macêdo Batista

IME - USP, 16 de Abril de 2013

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

await com semáforos: A técnica de passagem de bastão
Simplificação dos SIGNAL

await com
semáforos: A
técnica de
passagem de
bastão

Simplificação dos
SIGNAL

await com semáforos: A técnica de passagem de bastão

Segundo algoritmo (sem await)

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

```
int nr = 0, nw = 0;
# 0 <= (e+r+w) <=1 (semaforo binario dividido)
sem e = 1,
    r = 0,
    w = 0;
int dr = 0, dw = 0;

process Reader [i = 1 to M] {
    while (true) {
        # <await (nw == 0) nr = nr+1;>
        P(e);
        if (nw > 0) {dr = dr+1; V(e); P(r);}
        nr = nr+1;
        SIGNAL;
        le base de dados;
        # <nr = nr-1;>
        P(e);
        nr = nr-1; # Eh <await (true) ...;>
        SIGNAL;
    }
}
```

Segundo algoritmo (sem await)

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

```
process Writer [j = 1 to N] {  
    while (true) {  
        # <await (nr == 0 and nw == 0) nw = nw+1;>  
        P(e);  
        if (nr > 0 or nw > 0) {dw = dw+1; V(e); P(w);}  
        nw = nw+1;  
        SIGNAL;  
        escreve na base;  
        # <nw = nw-1;>  
        P(e);  
        nw = nw-1;  
        SIGNAL;  
    }  
}
```

SIGNAL

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

```
if (nw == 0 and dr > 0) {  
    dr = dr-1; V(r); # acorda um leitor, ou  
}  
elseif (nr == 0 and nw == 0 and dw > 0) {  
    dw = dw-1; V(w); # acorda um escritor, ou  
}  
else  
    V(e); /* Libera o lock da entrada */
```

- ☐ código usado para a saída de cada ação atômica
- ☐ sinaliza exatamente um dos 3 semáforos (e, r ou w)

SIGNAL

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

```
if (nw == 0 and dr > 0) {  
    dr = dr-1; V(r); # acorda um leitor, ou  
}  
elseif (nr == 0 and nw == 0 and dw > 0) {  
    dw = dw-1; V(w); # acorda um escritor, ou  
}  
else  
    V(e); /* Libera o lock da entrada */
```

- ☐ Algoritmo atual: tenta acordar primeiro leitores, depois escritores e depois qualquer processo querendo executar uma ação atômica
- ☐ Está dando prioridade para alguém?

Passagem de bastão

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

- ☐ O processo em execução é como se estivesse com o bastão
- ☐ Um processo ao sair da seção crítica chama o SIGNAL
- ☐ Ao chamar o SIGNAL, o processo passa o bastão para outros processos que estejam tentando executar
 - É o processo atual em execução que dá permissão para os outros executarem. Como se passasse o bastão

await com
semáforos: A técnica
de passagem de
bastão

▷ Simplificação dos
SIGNAL

Simplificação dos SIGNAL

SIGNAL executado antes de ler a base

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

- Geralmente os SIGNAL podem ser simplificados

```
# <await (nw == 0) nr = nr+1;>
P(e);
if (nw > 0) {dr = dr+1; V(e); P(r);}
nr = nr+1;
SIGNAL;
le base de dados;
```

- Com certeza nr é positivo e nw é zero

SIGNAL executado antes de ler a base

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

```
if (nw == 0 and dr > 0) {  
    dr = dr-1; V(r); # acorda um leitor, ou  
}  
elseif (nr == 0 and nw == 0 and dw > 0) {  
    dw = dw-1; V(w); # acorda um escritor, ou  
}  
else  
    V(e); /* Libera o lock da entrada */
```

- ☐ nr é positivo, então não precisa do elseif
- ☐ nw é zero, então não precisa comparar com zero no if

SIGNAL executado antes de ler a base

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

```
if (dr > 0) {  
    dr = dr-1; V(r); # acorda um leitor, ou  
}  
else  
    V(e); /* Libera o lock da entrada */
```

SIGNAL executado após ler a base

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

```
le base de dados;  
# <nr = nr-1;>  
P(e);  
nr = nr-1; # Eh <await (true) ...;>  
SIGNAL;
```

- ☐ Um leitor acabou de ler da base
- ☐ Como estão os valores de nw e dr neste ponto?

SIGNAL executado após ler a base

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

```
if (nw == 0 and dr > 0) {  
    dr = dr-1; V(r); # acorda um leitor, ou  
}  
elseif (nr == 0 and nw == 0 and dw > 0) {  
    dw = dw-1; V(w); # acorda um escritor, ou  
}  
else  
    V(e); /* Libera o lock da entrada */
```

- ☐ dr é zero, então não precisa do if
- ☐ nw é zero, então não precisa comparar com zero no elseif

SIGNAL executado após ler a base

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

```
if (nr == 0 and dw > 0) {  
    dw = dw-1; V(w); # acorda um escritor, ou  
}  
else  
    V(e); /* Libera o lock da entrada */
```

SIGNAL executado antes de escrever na base

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

```
# <await (nr == 0 and nw == 0) nw = nw+1;>
P(e);
if (nr > 0 or nw > 0) {dw = dw+1; V(e); P(w);}
nw = nw+1;
SIGNAL;
escreve na base;
```

- ☐ Escritor vai escrever na base
- ☐ Como estão os valores de nr e nw neste ponto?

SIGNAL executado antes de escrever na base

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

```
if (nw == 0 and dr > 0) {  
    dr = dr-1; V(r); # acorda um leitor, ou  
}  
elseif (nr == 0 and nw == 0 and dw > 0) {  
    dw = dw-1; V(w); # acorda um escritor, ou  
}  
else  
    V(e); /* Libera o lock da entrada */
```

- ☐ nr é zero, então não precisa comparar com zero no elseif
- ☐ mas nw é positivo, então não precisa do if nem do elseif

SIGNAL executado antes de escrever na base

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

```
V(e); /* Libera o lock da entrada */
```

SIGNAL executado após escrever na base

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

```
escreve na base;  
# <nw = nw-1;>  
P(e);  
nw = nw-1;  
SIGNAL;
```

- ☐ Escritor acabou de escrever na base
- ☐ Como estão os valores de nr e nw neste ponto?

SIGNAL executado após escrever na base

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

```
if (nw == 0 and dr > 0) {  
    dr = dr-1; V(r); # acorda um leitor, ou  
}  
elseif (nr == 0 and nw == 0 and dw > 0) {  
    dw = dw-1; V(w); # acorda um escritor, ou  
}  
else  
    V(e); /* Libera o lock da entrada */
```

- ☐ nr é zero, então não precisa comparar com zero no elseif
- ☐ nw é zero, então não precisa comparar com zero no if nem no elseif

SIGNAL executado após escrever na base

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

```
if (dr > 0) {  
    dr = dr-1; V(r); # acorda um leitor, ou  
}  
elseif (dw > 0) {  
    dw = dw-1; V(w); # acorda um escritor, ou  
}  
else  
    V(e); /* Libera o lock da entrada */
```

Leitores e escritores com SIGNAL simplificados

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

```
int nr = 0, nw = 0;
# 0 <= (e+r+w) <=1 (semaforo binario dividido)
sem e = 1,
    r = 0,
    w = 0;
int dr = 0, dw = 0;

process Reader [i = 1 to M] {
  while (true) {
    # <await (nw == 0) nr = nr+1;>
    P(e);
    if (nw > 0) {dr = dr+1; V(e); P(r);}
    nr = nr+1;
    if (dr > 0) {dr = dr-1; V(r);} else V(e);
    le base de dados;
    # <nr = nr-1;>
    P(e);
    nr = nr-1; # Eh <await (true) ...;>
    if (nr == 0 and dw > 0) {dw = dw-1; V(w);}
    else V(e);
  }
}
```

Leitores e escritores com SIGNAL simplificados

await com
semáforos: A técnica
de passagem de
bastão

Simplificação dos
SIGNAL

```
process Writer [j = 1 to N] {  
    while (true) {  
        # <await (nr == 0 and nw == 0) nw = nw+1;>  
        P(e);  
        if (nr > 0 or nw > 0) {dw = dw+1; V(e); P(w);}  
        nw = nw+1;  
        V(e);  
        escreve na base;  
        # <nw = nw-1;>  
        P(e);  
        nw = nw-1;  
        if (dr > 0) {dr = dr-1; V(r);}  
        elseif (dw > 0) {dw = dw-1; V(w);}  
        else V(e);  
    }  
}
```