

RELÁTÓRIO TP4

GUILHERME MARTINS SPECHT (20102977 - 4)

LUCAS DIAS NARDINO (21101052 - 5)

4646B-04 - Fundamentos de Sistemas Digitais - Turma 010 - 2021/2 - Prof. Rafael

Fraga Garibotti

ESPECIFICAÇÃO 3

PSEUDOCÓDIGO EM C++
main.cpp

```
#include <iostream>
using namespace std;
int main()
{
    int n = 6 , k = 0;
    int A[6] = {810, 100, 560, 380, 600, 87};
    int B[6] = {800, 555, 817, 124, 890, 456};
    int C[6] = {345, 200, 700, 180, 600, 490};
    int D[6];
    int S1 = 0, S2 = 0, S3 = 0, i;
    double MA, MB, MC;
```

} Vetores

```
for(i = 0; i < n; i++){
    S1 = S1 + A[i];
}
MA = S1/n;

for(i = 0; i < n; i++){
    S2 = S2 + B[i];
}
MB = S2/n;

for(i = 0; i < n; i++){
    S3 = S3 + C[i];
}
MC = S3/n;
```

} Calcula o valor das médias

```

if(MA < MB && MA < MC){
for(i = 0; i < n; i++){
if(MA < A[i]){
D[k] = A[i];
k++;
}
if(MA < B[i]){
D[k] = B[i];
k++;
}
if(MA < C[i]){
D[k] = C[i];
k++;
}
}
}
}

```

Analisa se a média A é a menor das três médias.

```

if(MB < MA && MB < MC){
for(i = 0; i < n; i++){
if(MB < A[i]){
D[k] = A[i];
k++;
}
if(MB < B[i]){
D[k] = B[i];
k++;
}
if(MB < C[i]){
D[k] = C[i];
k++;
}
}
}
}

```

Analisa se a média B é a menor das três médias.

```

if(MC < MB && MC < MA){
for(i = 0; i < n; i++){
if(MC < A[i]){
D[k] = A[i];
k++;
}
if(MC < B[i]){
D[k] = B[i];
k++;
}
if(MC < C[i]){
D[k] = C[i];
k++;
}
}
}
return 0}

```

Analisa se a média C é a menor das três médias.

Para a análise de C++ para Assembly consideraremos os laços “for” serem laços “while”.

Ex:

```
for(i = 0; i < n; i++){  
    S1 = S1 + A[i];  
}  
MA = S1/n;
```



```
i = 0;  
while(i < n){  
    S1 = S1 + A[i];  
    i++;  
}  
MA = S1/n;
```

CÓDIGO C++:

CÓDIGO MIPS:

int n; → \$s0

int k; → \$s7

int i; → \$t2

int A[]; → \$t3

int B[]; → \$t4

int C[]; → \$t5

int D[]; → \$t0

double MA ;
double MB; } double MV; → \$t9
double MC;

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	4	0	255	801	198	433	100	200
0x10010020	300	400	450	666	20	780	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0

0x10010000 (.data) ☒ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

■ n = Número de elementos dos vetores A, B e C.

■ k = Número de elementos do vetor D.

■ Vetor A

■ Vetor B

■ Vetor C

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	4	8	255	801	198	433	100	200
0x10010020	300	400	450	666	20	780	255	450
0x10010040	801	666	300	433	400	780	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0

0x10010000 (.data) ▼ ☒ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

 n = Número de elementos dos vetores A, B e C.

 k = Número de elementos do vetor D.

 Vetor A

 Vetor B

 Vetor C

 Vetor D

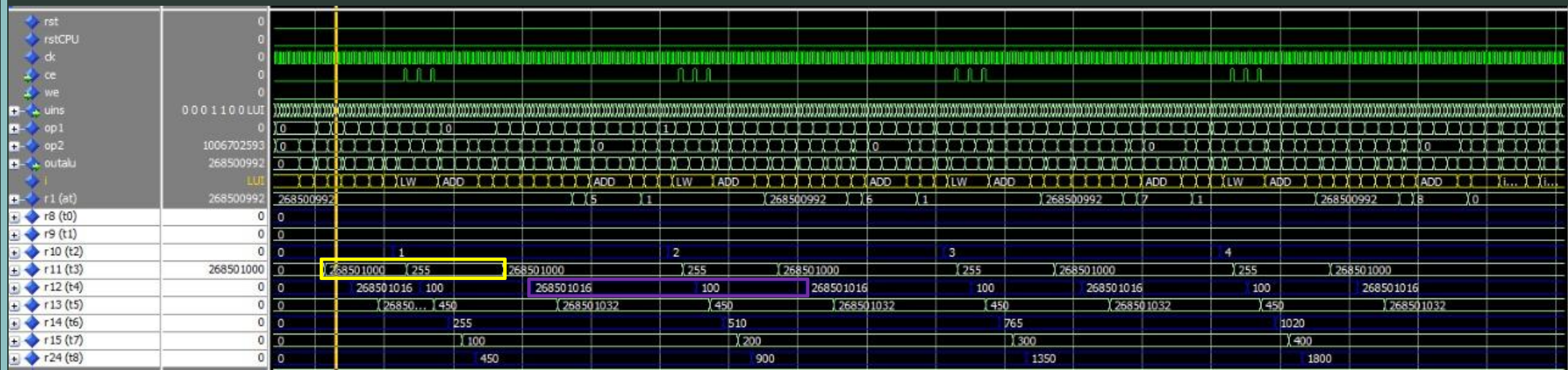
mem0	4	4
mem1	0	0
mem2	255	255
mem3	801	801
mem4	198	198
mem5	433	433
mem6	100	100
mem7	200	200
mem8	300	300
mem9	400	400
mem10	450	450
mem11	666	666
mem12	20	20
mem13	780	780
mem14	0	0
mem15	0	0
mem16	0	0
mem17	0	0
mem18	0	0
mem19	0	0

.data

```

n: .word 4
k: .word 0
A: .word 255 801 198 433
B: .word 100 200 300 400
C: .word 450 666 20 780
D: .word 0

```



- Leitura do Primeiro Vetor e Média 1
- Leitura do Segundo Vetor e Média 2

◆ mem0	4	4	
◆ mem1	0	0	
◆ mem2	255	255	
◆ mem3	801	801	
◆ mem4	198	198	
◆ mem5	433	433	
◆ mem6	100	100	
◆ mem7	200	200	
◆ mem8	300	300	
◆ mem9	400	400	
◆ mem10	450	450	
◆ mem11	666	666	
◆ mem12	20	20	
◆ mem13	780	780	
◆ mem14	450	450	
◆ mem15	450	450	
◆ mem16	450	0 450	
◆ mem17	450	0 450	
◆ mem18	450	0 450	
◆ mem19	450	0 450	

Final da Simulação