PSI3441 – Arquitetura de Sistemas Embarcados

- Assembly vs C

Escola Politécnica da Universidade de São Paulo

Prof. Gustavo Rehder – gprehder@usp.br

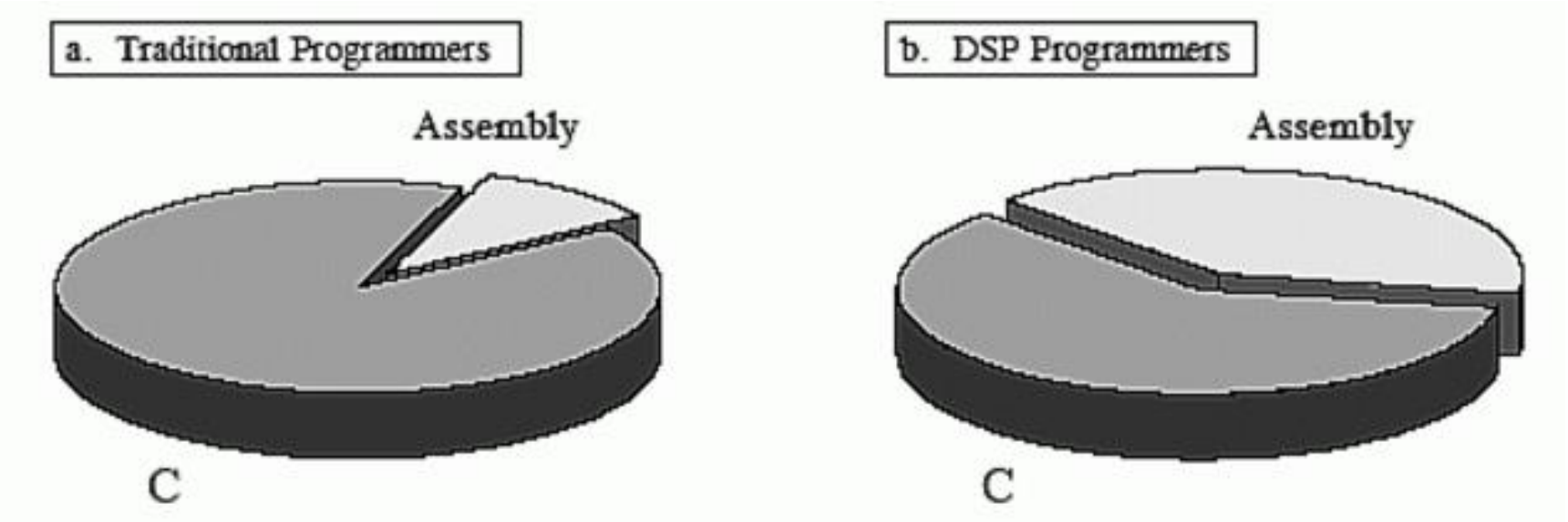
2020

Pergunta

1. Em qual linguagem programar microcontroladores?
   - C (mais usado) e C++ (menos usado)
   - Assembly
2. Qual é a diferença entre C e Assembly?
   - Nível
   - Dificuldade/Esforço/Tempo
   - Eficiência
   - Necessidade de conhecimento do Hardware

Mais nos próximos slides...

Quando usar Assembly?



DSP – Digital Signal Processor – Processadores otimizados para processamento de sinais de áudio e vídeo em tempo real.

Produto Escalar em C

Produto Escalar

$$\boldsymbol{x}[n] \cdot \boldsymbol{y}[n] \rightarrow x[1] \cdot y[1] + x[2] \cdot y[2] + \cdots + x[n] \cdot y[n]$$

```
001   #define LEN 20
002    float dm x[LEN];
003    float pm y[LEN];
004    float result;
005
006   main()
007
008   {
009    int n;
010    float s;
011    for (n=0;n<LEN;n++)
012      s += x[n]*y[n];
013    result = s
014   }
```

Disassembly do Código em C compilado pelo GCC

```
push {r7,lr}
sub sp,sp,#8
add r7,sp,#0
 movs r3,#0
str r3,[r7,#4]
b main+0x6 (0xfb2)
ldr r3,[r7,#4]
adds r3,#1
str r3,[r7,#4]
ldr r3,[r7,#4]
cmp r3,#19
ble main+0x6 (0xfac)
ldr r3,[pc,#44]
ldr r2,[r7,#4]
lsls r2,r2,#2
ldr r2,[r2,r3]
ldr r3,[pc,#40]
```

```
ldr r1,[r7,#4]
  lsls r1,r1,#2
  ldr r3,[r1,r3]
  mov r0,r2
  mov r1,r3
  bl __aeabi_fmul (0xc6c)
  mov r3,r0
  ldr r0,[r7,#0]
  mov r1,r3
  bl __aeabi_fadd (0x8c8)
  mov r3,r0
  str r3,[r7,#0]
  ldr r3,[pc,#16]
  ldr r2,[r7,#0]
  str r2,[r3,#0]
```

Produto Escalar em Assembly

para Analog Device SHARC DSP

±1 ciclo de clock por linha

```
001   i12 = _y;                          /* i12 points to beginning of y[ ] */
002   i4 = _x;                           /* i4 points to beginning of x[ ] */
003
004   lcntr = 20, do (pc,4) until lce;   /* loop for the 20 array entries */
005     f2 = dm(i4,m6);                  /* load the x[ ] value into register f2 */
006     f4 = pm(i12,m14);                /* load the y[ ] value into register f4 */
007     f8 = f2*f4;                      /* multiply the two values, store in f8 */
008     f12 = f8 + f12;                  /* add the product to the accumulator in f12 */
009
010   dm(_result) = f12;                 /* write the accumulator to memory */
```

4 ciclos

Código Otimizado

```
001   i12 = _y;                                                    /* i12 points to beginning of y[ ] */
002   i4 = _x;                                                     /* i4 points to beginning of x[ ] */
003
004   f2 = dm(i4,m6), f4 = pm(i12,m14)                             /* prime the registers */
005   f8 = f2*f4, f2 = dm(i4,m6), f4 = pm(i12,m14);
006
007   lcntr = 18, do (pc,1) until lce;                             /* highly efficient main loop */
008   f12 = f8 + f12, f8 = f2*f4, f2 = dm(i4,m6), f4 = pm(i12,m14);
009
010   f12 = f8 + f12, f8 = f2*f4;                                  /* complete the last loop */
011   f12 = f8 + f12;
012
013   dm(_result) = f12;                                           /* store the result in memory */
```

1 ciclo