- Declaração de variáveis

```
1  int main(){
2      int a = 4;
3  }
```

```
                    main:
40048255            push rbp
40048348 89 e5      mov rbp,rsp
400486c7 45 fc 04 00 00 0  mov DWORD PTR [rbp-0x4],0x4
40048db8 00 00 00 00       mov eax,0x0
4004925d            pop rbp
400493c3            ret
40049466 2e 0f 1f 84 00 0  nop WORD PTR cs:[rax+rax*1+0x0]
40049e66 90         xchg ax,ax
```

- Estrutura condicional simples

```
1  int main(){
2      int a = 4;
3      if (a > 4){
4          a = 5;
5      }
6  }
```

```
40048255            push rbp
40048348 89 e5      mov rbp,rsp
400486c7 45 fc 04 00 00 0  mov DWORD PTR [rbp-0x4],0x4
40048d83 7d fc 04   cmp DWORD PTR [rbp-0x4],0x4
4004917e 07         jle 40049a <main+0x18>
400493c7 45 fc 05 00 00 0  mov DWORD PTR [rbp-0x4],0x5
40049ab8 00 00 00 00       mov eax,0x0
```

```
1  int main(){
2      int a = 4;
3      if (a > 4 && a < 6){
4          a = 5;
5      }
6  }
```

```
400486c7 45 fc 04 00 00 0  mov DWORD PTR [rbp-0x4],0x4
40048d83 7d fc 04   cmp DWORD PTR [rbp-0x4],0x4
4004917e 0d         jle 4004a0 <main+0x1e>
40049383 7d fc 05   cmp DWORD PTR [rbp-0x4],0x5
4004977f 07         jg 4004a0 <main+0x1e>
400499c7 45 fc 05 00 00 0  mov DWORD PTR [rbp-0x4],0x5
4004a0b8 00 00 00 00       mov eax,0x0
```

- Estrutura condicional composta

```
1  int main(){
2      int a = 4;
3      if (a > 4){
4          a = 5;
5      }
6      else if (a < 4){
7          a = 3;
8      }
9      else{
10         a = 0;
11     }
12 }
```

```
400486c7 45 fc 04 00 00 0  mov DWORD PTR [rbp-0x4],0x4
40048d83 7d fc 04   cmp DWORD PTR [rbp-0x4],0x4
4004917e 09         jle 40049c <main+0x1a>
400493c7 45 fc 05 00 00 0  mov DWORD PTR [rbp-0x4],0x5
40049aeb 16         jmp 4004b2 <main+0x30>
40049c83 7d fc 03   cmp DWORD PTR [rbp-0x4],0x3
4004a07f 09         jg 4004ab <main+0x29>
4004a2c7 45 fc 03 00 00 0  mov DWORD PTR [rbp-0x4],0x3
4004a9eb 07         jmp 4004b2 <main+0x30>
4004abc7 45 fc 00 00 00 0  mov DWORD PTR [rbp-0x4],0x0
4004b2b8 00 00 00 00       mov eax,0x0
```

- Estrutura de repetição FOR

```
1  int main(){
2      int a;
3      int b = 0;
4      for (a=0; a<4; a++){
5          b = b + a;
6      }
7  }
```

```
400486c7 45 f8 00 00 00 0  mov DWORD PTR [rbp-0x8],0x0
40048dc7 45 fc 00 00 00 0  mov DWORD PTR [rbp-0x4],0x0
400494eb 0a                jmp 4004a0 <main+0x1e>
4004968b 45 fc             mov eax,DWORD PTR [rbp-0x4]
40049901 45 f8             add DWORD PTR [rbp-0x8],eax
40049c83 45 fc 01          add DWORD PTR [rbp-0x4],0x1
4004a083 7d fc 03          cmp DWORD PTR [rbp-0x4],0x3
4004a47e f0                jle 400496 <main+0x14>
4004a6b8 00 00 00 00       mov eax,0x0
```

- Estrutura de repetição WHILE

```
1  int main(){
2      int a = 5;
3      while(a > 0){
4          a = a - 1;
5      }
6  }
```

```
400486c7 45 fc 05 00 00 0  mov DWORD PTR [rbp-0x4],0x5
40048deb 04                jmp 400493 <main+0x11>
40048f83 6d fc 01          sub DWORD PTR [rbp-0x4],0x1
40049383 7d fc 00          cmp DWORD PTR [rbp-0x4],0x0
4004977f f6                jg 40048f <main+0xd>
400499b8 00 00 00 00       mov eax,0x0
```

- Vetor

```
1  int main(){
2      int a = 4;
3      int b[5];
4      b[0] = 10;
5      b[1] = 20;
6      b[2] = 30;
7      b[3] = 40;
8      b[4] = 50;
9      b[a] = 51;
10 }
```

```
400486c7 45 fc 04 00 00 0  mov DWORD PTR [rbp-0x4],0x4
40048dc7 45 e0 0a 00 00 0  mov DWORD PTR [rbp-0x20],0xa
400494c7 45 e4 14 00 00 0  mov DWORD PTR [rbp-0x1c],0x14
40049bc7 45 e8 1e 00 00 0  mov DWORD PTR [rbp-0x18],0x1e
4004a2c7 45 ec 28 00 00 0  mov DWORD PTR [rbp-0x14],0x28
4004a9c7 45 f0 32 00 00 0  mov DWORD PTR [rbp-0x10],0x32
4004b08b 45 fc             mov eax,DWORD PTR [rbp-0x4]
4004b348 98                cdqe
4004b5c7 44 85 e0 33 00 0  mov DWORD PTR [rbp+rax*4-0x20],0x33
4004bdb8 00 00 00 00       mov eax,0x0
```

```
1  int main(){
2      int a;
3      int b[5];
4      for (a=0; a<5; a++){
5          b[a] = a+10;
6      }
7  }
```

```
400486c7 45 fc 00 00 00 0  mov DWORD PTR [rbp-0x4],0x0
40048deb 13                jmp 4004a2 <main+0x20>
40048f8b 45 fc             mov eax,DWORD PTR [rbp-0x4]
4004928d 50 0a             lea edx,[rax+0xa]
4004958b 45 fc             mov eax,DWORD PTR [rbp-0x4]
40049848 98                cdqe
40049a89 54 85 e0          mov DWORD PTR [rbp+rax*4-0x20],edx
40049e83 45 fc 01          add DWORD PTR [rbp-0x4],0x1
4004a283 7d fc 04          cmp DWORD PTR [rbp-0x4],0x4
4004a67e e7                jle 40048f <main+0xd>
4004a8b8 00 00 00 00       mov eax,0x0
```

- Matriz

```
1   int main(){
2       int a;
3       int b[3][3];
4       b[0][0] = 10;
5       b[0][1] = 20;
6       b[0][2] = 30;
7       b[1][0] = 10;
8       b[1][1] = 20;
9       b[1][2] = 30;
10      b[2][0] = 10;
11      b[2][1] = 20;
12      b[2][2] = 30;
13  }
```

```
400486c7 45 d0 0a 00 00 0  mov DWORD PTR [rbp-0x30],0xa
40048dc7 45 d4 14 00 00 0  mov DWORD PTR [rbp-0x2c],0x14
400494c7 45 d8 1e 00 00 0  mov DWORD PTR [rbp-0x28],0x1e
40049bc7 45 dc 0a 00 00 0  mov DWORD PTR [rbp-0x24],0xa
4004a2c7 45 e0 14 00 00 0  mov DWORD PTR [rbp-0x20],0x14
4004a9c7 45 e4 1e 00 00 0  mov DWORD PTR [rbp-0x1c],0x1e
4004b0c7 45 e8 0a 00 00 0  mov DWORD PTR [rbp-0x18],0xa
4004b7c7 45 ec 14 00 00 0  mov DWORD PTR [rbp-0x14],0x14
4004bec7 45 f0 1e 00 00 0  mov DWORD PTR [rbp-0x10],0x1e
4004c5b8 00 00 00 00       mov eax,0x0
```

- Declaração de registros

```
1   struct coordenada{
2       int x,y;
3   };
4
5   int main(){
6       struct coordenada a;
7       a.x = 10;
8       a.y = 30;
9   }
```

```
40048255                   push rbp
40048348 89 e5             mov rbp,rsp
400486c7 45 f8 0a 00 00 0  mov DWORD PTR [rbp-0x8],0xa
40048dc7 45 fc 1e 00 00 0  mov DWORD PTR [rbp-0x4],0x1e
400494b8 00 00 00 00       mov eax,0x0
```

- Manipulando arquivos

```
1   #include <stdio.h>
2
3   int main(){
4       FILE *arq;
5       arq = fopen("arquivo.txt","r");
6       fputc('a',arq);
7       fclose(arq);
8   }
```

```
                           fclose@plt:
400470ff 25 a2 0b 20 00    jmp QWORD PTR [rip+0x200ba2] # 601018 <fclose@GLIBC_2.2.5>
40047668 00 00 00 00       push 0x0
40047be9 e0 ff ff ff       jmp 400460 <.plt>
                           fputc@plt:
40048 0ff 25 9a 0b 20 00   jmp QWORD PTR [rip+0x200b9a] # 601020 <fputc@GLIBC_2.2.5>
40048668 01 00 00 00       push 0x1
40048be9 d0 ff ff ff       jmp 400460 <.plt>
                           fopen@plt:
40049 0ff 25 92 0b 20 00   jmp QWORD PTR [rip+0x200b92] # 601028 <fopen@GLIBC_2.2.5>
40049668 02 00 00 00       push 0x2
40049be9 c0 ff ff ff       jmp 400460 <.plt>
```

```
40058abe 54 06 40 00       mov esi,0x400654
40058fbf 56 06 40 00       mov edi,0x400656
400594e8 f7 fe ff ff       call 400490 <fopen@plt>
40059948 89 45 f8          mov QWORD PTR [rbp-0x8],rax
40059d48 8b 45 f8          mov rax,QWORD PTR [rbp-0x8]
4005a148 89 c6             mov rsi,rax
4005a4bf 61 00 00 00       mov edi,0x61
4005a9e8 d2 fe ff ff       call 400480 <fputc@plt>
4005ae48 8b 45 f8          mov rax,QWORD PTR [rbp-0x8]
4005b248 89 c7             mov rdi,rax
4005b5e8 b6 fe ff ff       call 400470 <fclose@plt>
4005bab8 00 00 00 00       mov eax,0x0
```

- Sub-rotinas (funções)

```
1   int soma(int a, int b){
2       int c = a + b;
3       return c;
4   }
5
6   int main(){
7       int a = 2;
8       int b = 3;
9       soma(a,b);
10  }
```

```
                    soma:
40048255            push rbp
40048348 89 e5      mov rbp,rsp
40048689 7d ec      mov DWORD PTR [rbp-0x14],edi
40048989 75 e8      mov DWORD PTR [rbp-0x18],esi
40048c8b 55 ec      mov edx,DWORD PTR [rbp-0x14]
40048f8b 45 e8      mov eax,DWORD PTR [rbp-0x18]
40049201 d0         add eax,edx
40049489 45 fc      mov DWORD PTR [rbp-0x4],eax
4004978b 45 fc      mov eax,DWORD PTR [rbp-0x4]
40049a5d            pop rbp
40049bc3            ret
                    main:
40049c55            push rbp
40049d48 89 e5      mov rbp,rsp
4004a048 83 ec 10   sub rsp,0x10
4004a4c7 45 fc 02 00 00 0  mov DWORD PTR [rbp-0x4],0x2
4004abc7 45 f8 03 00 00 0  mov DWORD PTR [rbp-0x8],0x3
4004b28b 55 f8      mov edx,DWORD PTR [rbp-0x8]
4004b58b 45 fc      mov eax,DWORD PTR [rbp-0x4]
4004b889 d6         mov esi,edx
4004ba89 c7         mov edi,eax
4004bce8 c1 ff ff ff  call 400482 <soma>
4004c1b8 00 00 00 00  mov eax,0x0
```