Linguagem C

Uma revisão com foco em sistemas embarcados

Marcelo Barros

UFU/FEELT

21 de setembro de 2022

Outline

Definições

- Revisão C11 (ISO/IEC 9899:2011)
- Compilador GNU GCC

Linha do tempo da linguagem C

- B (1972): Primeira implementação, Dennis Ritchie and Ken Thompson e colegas para o PDP11.
- K&R (1978): Primeira especificação informal.
- C89/C90 (1989/1990): Adoção como padrão pela ANSI (C89) e depois pela ISO (C90).
- C99 (1999): Primeira grande revisão do padrão, amplamente utilizada.
- C11 (2011): Segunda revisão da linguagem. Aproximação ao C++. Ainda em uso moderado.
- C17 (2018): Sem características novas. Apenas correções.
- C2x (2023?)[1]

[1] https://en.wikipedia.org/wiki/C2x

Existe mesmo uma linguagem "Embedded C" ?

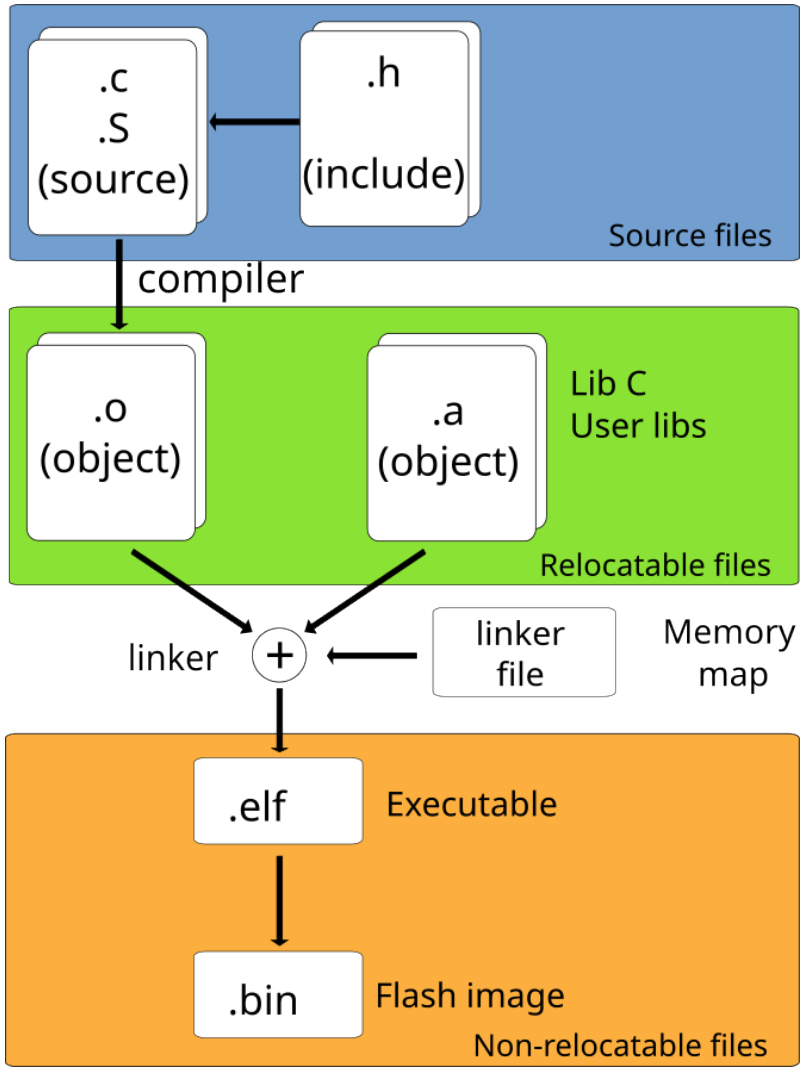
- Norma: *Programming languages — C — Extensions to support embedded processors* ISO/IEC TR 18037:2008
- Tecnicamente é apenas uma extensão do C, cobrindo:
  - Aritmética de ponto fixo
  - Espaço de endereçamento
  - Endereçamento de hardware básico para I/O

Palavras Reservadas

- Não devem ser usadas como nomes de funções ou variáveis
- São *case sensitives*
- Em azul, as adições do C11 em relação ao C99

```
auto break case char const continue default
do double else enum extern float for goto
if inline int long register restrict return
short signed sizeof static struct switch
typedef union unsigned void volatile while
_Bool _Complex _Imaginary
_Alignas _Alignof _Atomic _Generic
_Noreturn _Static_assert _Thread_local
```

Estrutura de uma aplicação



Arquivos de inclusão

```c
#ifndef __DEMO_H__
#define __DEMO_H__

#ifdef __cplusplus
extern "C" {
#endif

// defines

// tipos de dados compartilhados
// (estruturas, unioes, typedefs)

// prototipo de funcoes

#ifdef __cplusplus
}
#endif

#endif /* __DEMO_H__ */
```

- Pense como API: só exporte interfaces e o que elas precisarem !
- Não esqueça a proteção de inclusão recursiva !
- Não é ANSI-C mas `#pragma once` pode ser interessante
- Evite incluir arquivos de inclusão como boa prática

Arquivos de inclusão

```c
#ifndef __ACCEL_H__
#define __ACCEL_H__

#ifdef __cplusplus
extern "C" {
#endif

#define ACCEL_NUM_AXIS 3

typedef struct accel_data_s
{
  float axis[ACCEL_NUM_AXIS];
} accel_data_t;

void accel_read(accel_data_t *data);
void accel_init(void);

#ifdef __cplusplus
}
#endif

#endif /* __ACCEL_H__ */
```

Arquivos de código fonte

```
// inclusoes da lib c

// inclusoes do projeto

// defines internos

// tipos de dados internas
// (estruturas, unioes, typedefs)

// constante internas (const)

// variaveis internas (static)

// prototipos de funcoes internas

// funcoes internas (static)
// ou exportadas (ver .h)
```

- Externe via funções o acesso a seus dados internos (encapsulamento)
- Dê escopo de arquivo para as suas funções e variáveis com static !
- Salve RAM colocando como const o que for realmente constante.

Arquivos de código fonte

```c
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>

#include "demo.h"

static bool started = false;

static void accel_init_drv(void)
{
}

void accel_read(accel_data_t *data)
{
}

void accel_init(void)
{
  if(!started)
  {
    accel_init_drv();
    started = true;
  }
}
```

```c
// test
#include <stdio.h>
int main(void)
{
  printf("Hello World!");
}
```

Palavras Reservadas

- x