Bootloaders

Teoría e implementación

Mg. Ing. Gonzalo Nahuel Vaca

AADECA

14 de noviembre de 2024

Contenido

Introducción

¿Qué es un bootloader?

BOOT - Momento de ejecución

ARRANQUE

LOADER - Funcionalidad

CARGA OTRO PROGRAMA

¿Para que sirve un bootloader?

Root of Trust

Punto de confianza inicial en un esquema de ciberseguridad.

Verificar integridad

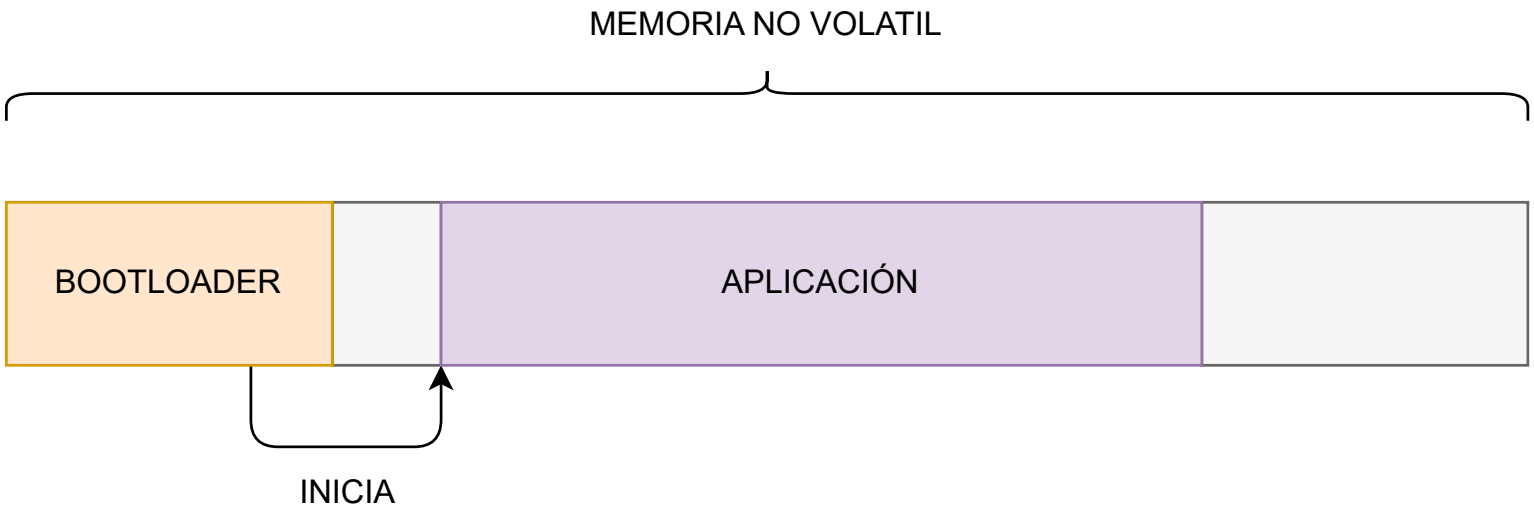
Puede validar la firma del programa a ejecutar.

Actualizaciones

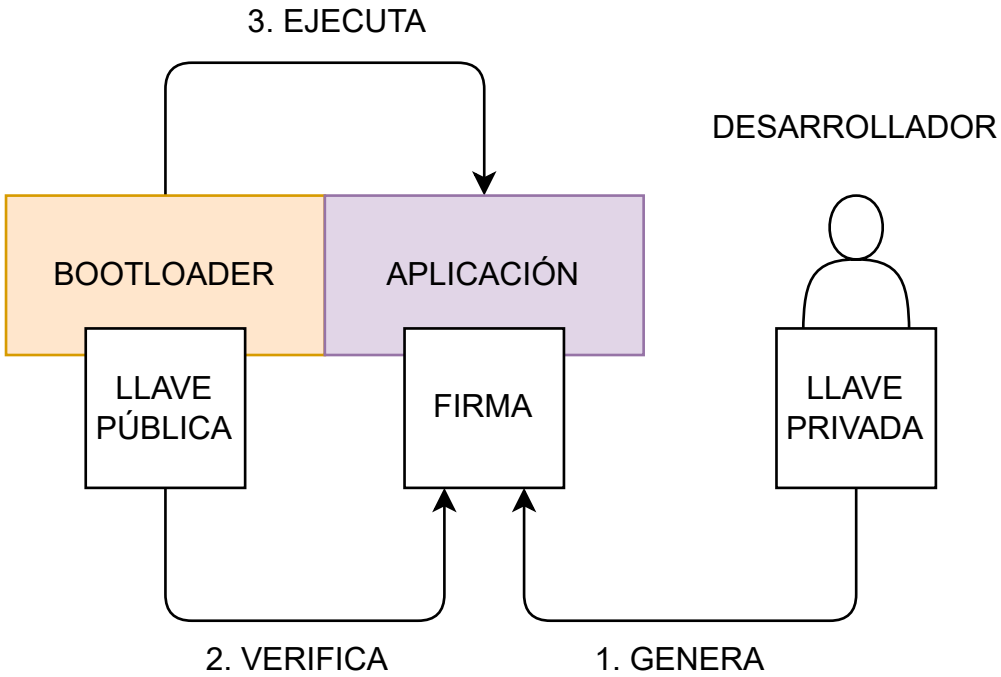
Ofrece *Serial Recovery* y permite actualizaciones *Over The Air*

Teoría de funcionamiento

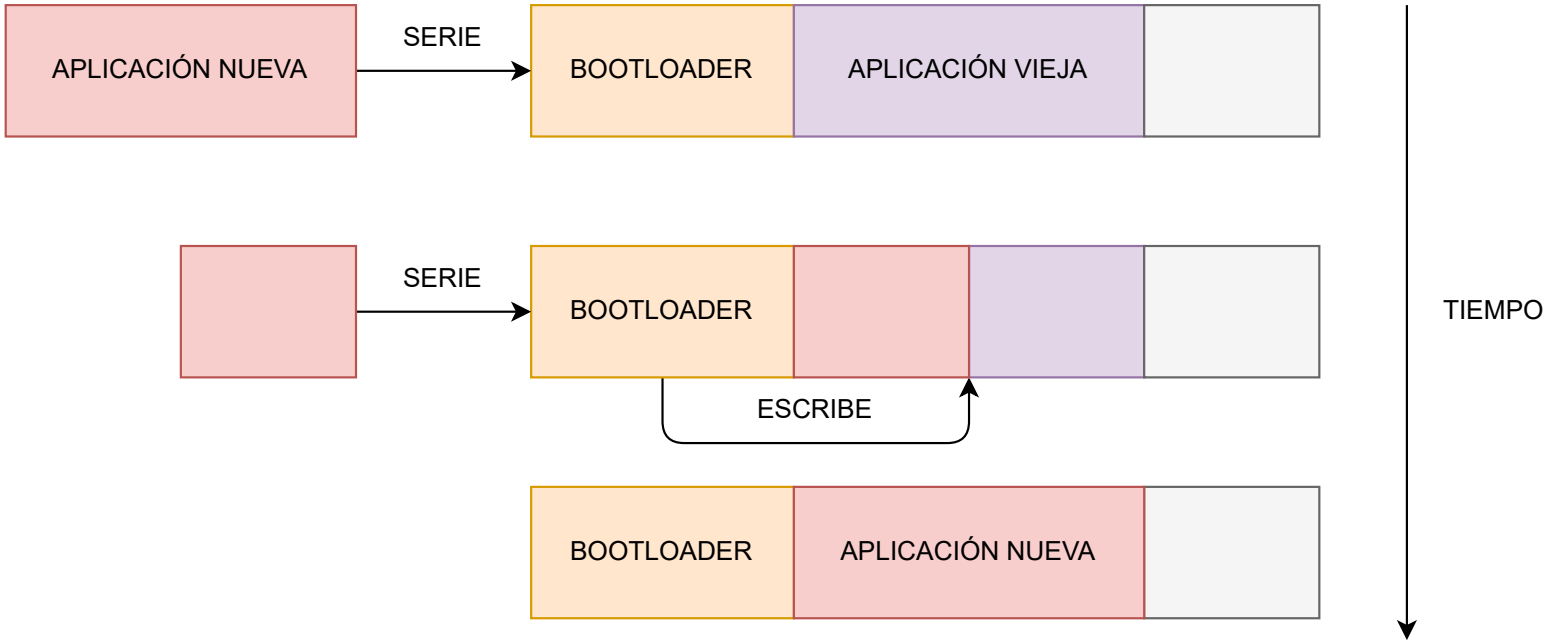
Funcionamiento básico



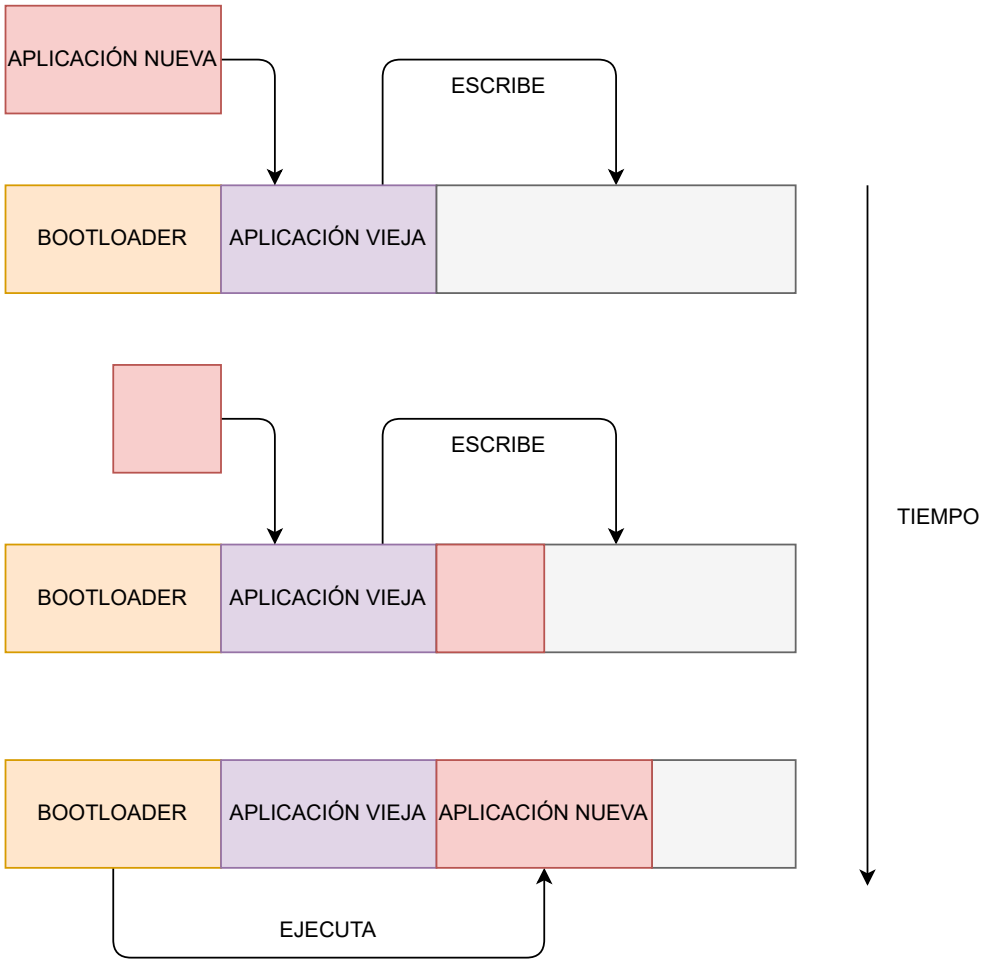
Root of Trust



Serial Recovery



Firmware Over The Air (FOTA) Update



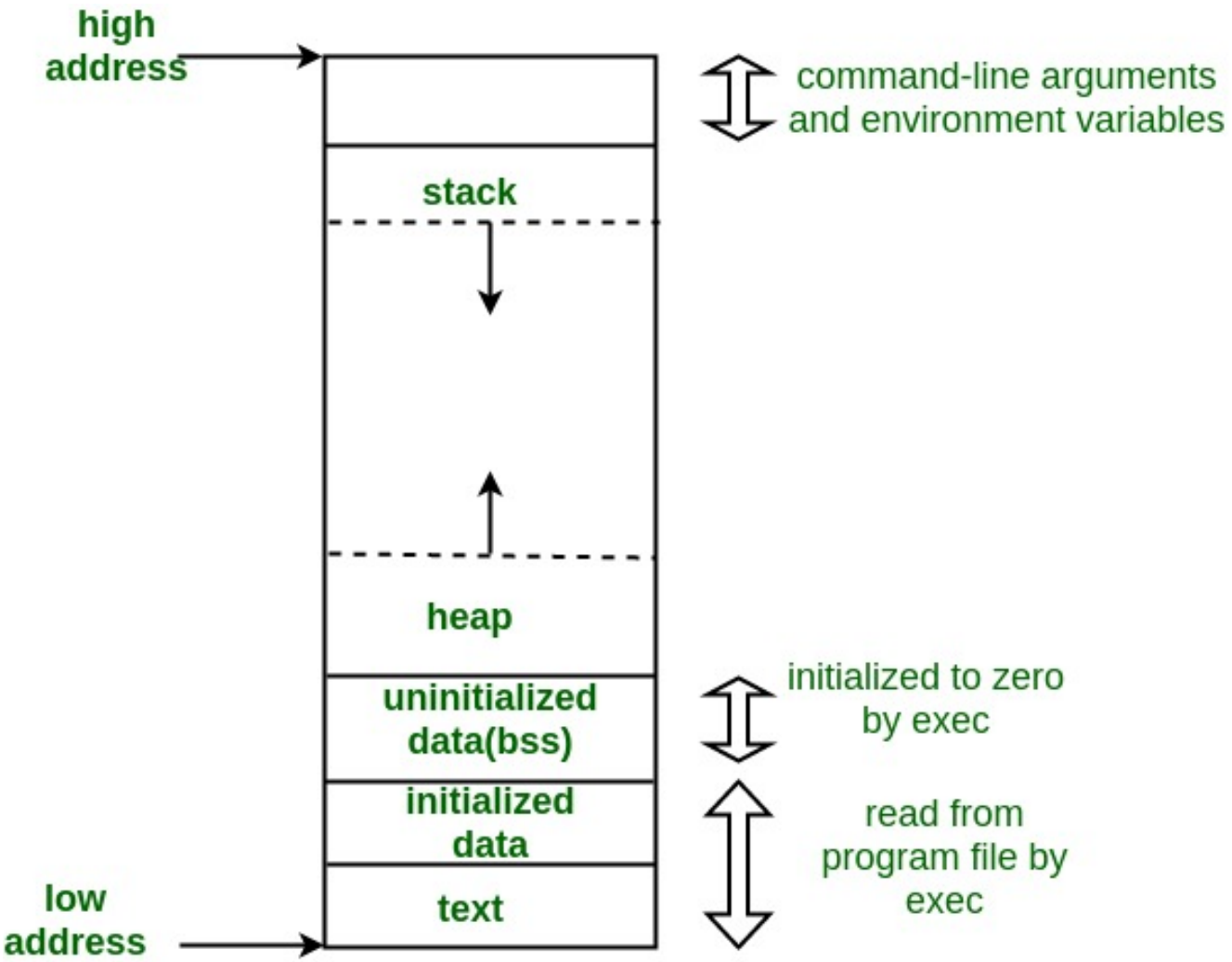
Implementación

Arquitectura

Aclaración

Implementación válida para microcontroladores ARM Cortex M.

Modelo de memoria



Vector Table Offset Register

```
__Vectors       DCD     __initial_sp              ; Top of Stack initialization
                DCD     Reset_Handler             ; Reset Handler
                DCD     NMI_Handler               ; NMI Handler
                DCD     HardFault_Handler         ; Hard Fault Handler
                DCD     MemManage_Handler         ; MPU Fault Handler
                DCD     BusFault_Handler          ; Bus Fault Handler
                DCD     UsageFault_Handler        ; Usage Fault Handler
                DCD     SecureFault_Handler       ; Secure Fault Handler
                DCD     0                         ; Reserved
                DCD     0                         ; Reserved
                DCD     0                         ; Reserved
                DCD     SVC_Handler               ; SVC Handler
                DCD     DebugMon_Handler          ; Debug Monitor Handler
                DCD     0                         ; Reserved
                DCD     PendSV_Handler            ; PendSV Handler
                DCD     SysTick_Handler           ; SysTick Handler
```

CMSIS GCC - Start

```
1  __STATIC_FORCEINLINE __NO_RETURN void __cmsis_start(
       void)
2  {
3      extern void _start(void) __NO_RETURN;
4
5      /* ... */
6
7      extern const __copy_table_t __copy_table_start__;
8      extern const __copy_table_t __copy_table_end__;
9      extern const __zero_table_t __zero_table_start__;
10     extern const __zero_table_t __zero_table_end__;
11
12     /* ... */
13 }
```

CMSIS GCC - Atributo VECTOR TABLE

```
1 #ifndef __VECTOR_TABLE
2 #define __VECTOR_TABLE                  __Vectors
3 #endif
4
5 #ifndef __VECTOR_TABLE_ATTRIBUTE
6 #define __VECTOR_TABLE_ATTRIBUTE   __attribute__((used,
      section(".vectors")))
7 #endif
```

Saltar a la aplicación

```
1  #include <stdint.h>
2  #include "cmsis_gcc.h"
3
4  __attribute__((noreturn)) void jump_to_application(
       uint32_t app_address)
5  {
6      uint32_t msp = *(uint32_t *)(app_address);
7      uint32_t reset_vector = *(uint32_t *)(app_address
           + 4U);
8      __set_MSP(msp); // __ASM volatile ("MSR msp, %0" :
           : "r" (topOfMainStack) : );
9      asm volatile("bx %0" : : "r"(reset_vector));
10     while(1);
11 }
```

Makefile - Toolchain y Flags

```
1  PREFIX ?= arm-none-eabi-
2
3  AR      := $(PREFIX)ar
4  AS      := $(PREFIX)as
5  CC      := $(PREFIX)gcc
6  CXX     := $(PREFIX)g++
7  GDB     := $(PREFIX)gdb
8  LD      := $(PREFIX)gcc
9  OBJCOPY := $(PREFIX)objcopy
10 OBJDUMP := $(PREFIX)objdump
11
12 CFLAGS = -mthumb -mcpu=cortex-m4
```

Makefile - Targets

```
1  snipets.objdump: snipets.o
2      @$(OBJDUMP) -d $< > $@
3
4  snipets.o: snipets.c
5      $(CC) $(CFLAGS) -c $< -o $@
6
7  clean:
8      -rm snipets.o snipets.objdump
9
10 .PHONY: clean
```

Tecnologías disponibles y uso real

Zephyr RTOS + MCUboot

Eclipse hawkBit

Ejemplo real: Pump Control

Desafíos

- Disperción geográfica (Egipto, India, LATAM, etc.).
- Robo de propiedad intelectual.
- Sistemas críticos.

Soluciones

- Bootloaders propios.
- Programadores de producción y de *calle*.
- Sistema de gestión y despliegue de firmware propio.

Gracias por su atención

Contacto: vacagonzalo@gmail.com
LinkedIn: linkedin.com/in/vaca-gonzalo
Github: github.com/vacagonzalo/aadeca-bootloaders-2024