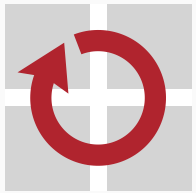
StuBSπ

Multicore Boot für den Raspberry Pi

2020-03-13

Daniel Bauer Sven Leykauf

Friedrich-Alexander-Universität Erlangen-Nürnberg



Lehrstuhl für Verteilte Systeme
und Betriebssysteme

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Erinnerung: Aufgabenstellung

- StuBSπ: Studenten Betriebssystem auf Raspberry Pi
- Voll funktionsfähig unter QEMU
- Problem: Nur ein Kern auf echter Hardware

Erinnerung: Aufgabenstellung

- StuBSπ: Studenten Betriebssystem auf Raspberry Pi
- Voll funktionsfähig unter QEMU
- Problem: Nur ein Kern auf echter Hardware

Aufgabe

Entwicklung von Multicore Boot-Code für den Raspberry Pi 2 in StuBSπ.

Bootprozess

Hinweis

Raspberry Pi besitzt einen sehr “einzigartigen” Bootprozess.

Bootprozess

Hinweis

Raspberry Pi besitzt einen sehr “einzigartigen” Bootprozess.

- GPU: Feststellen der Bootmediums
- GPU: Laden von `bootcode.bin`
- GPU: Laden von `start.elf`
- GPU: Parsen von `config.txt`
- GPU: Laden des Kernelimages `kernel7/8.img`

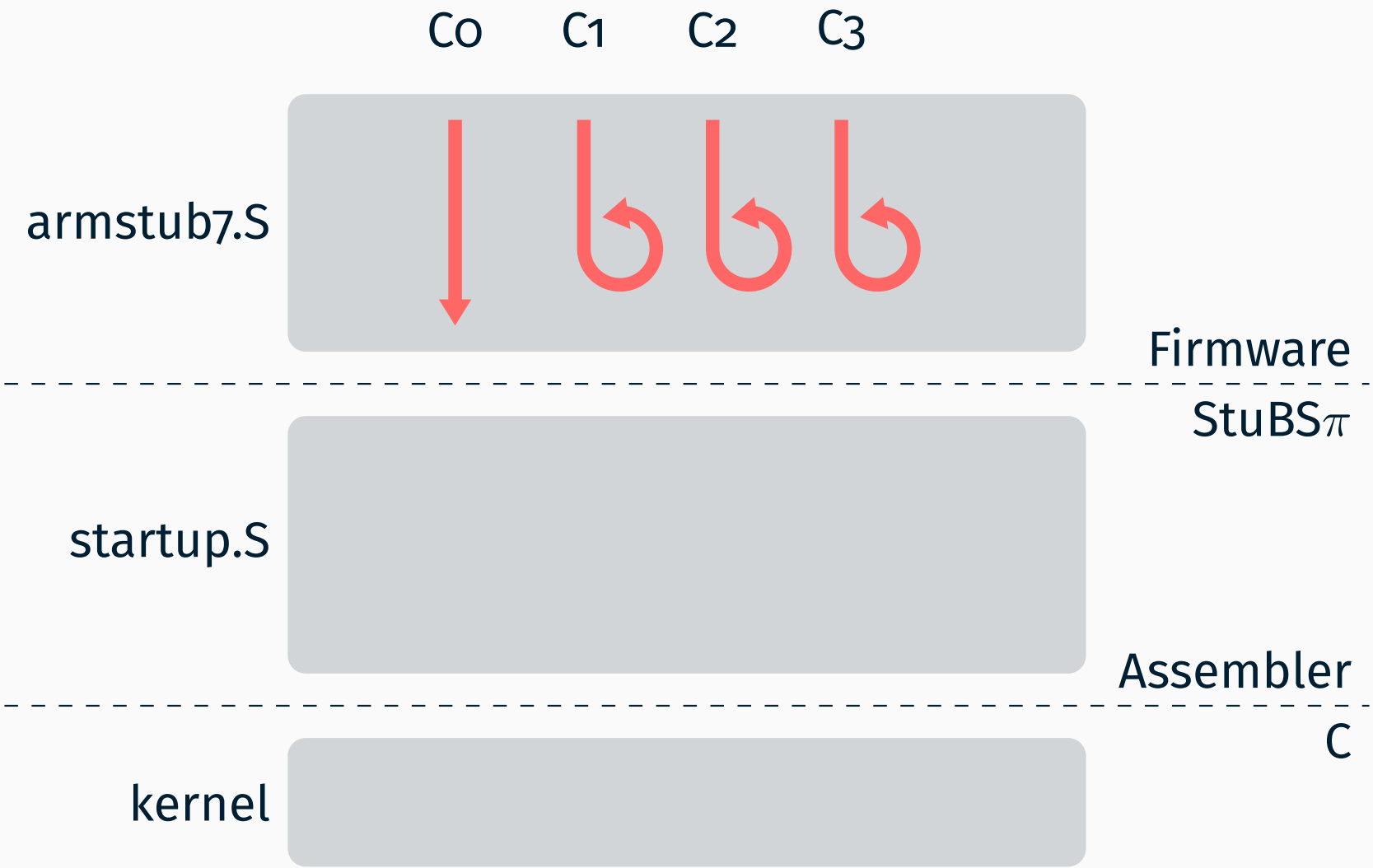
Bootprozess

Hinweis

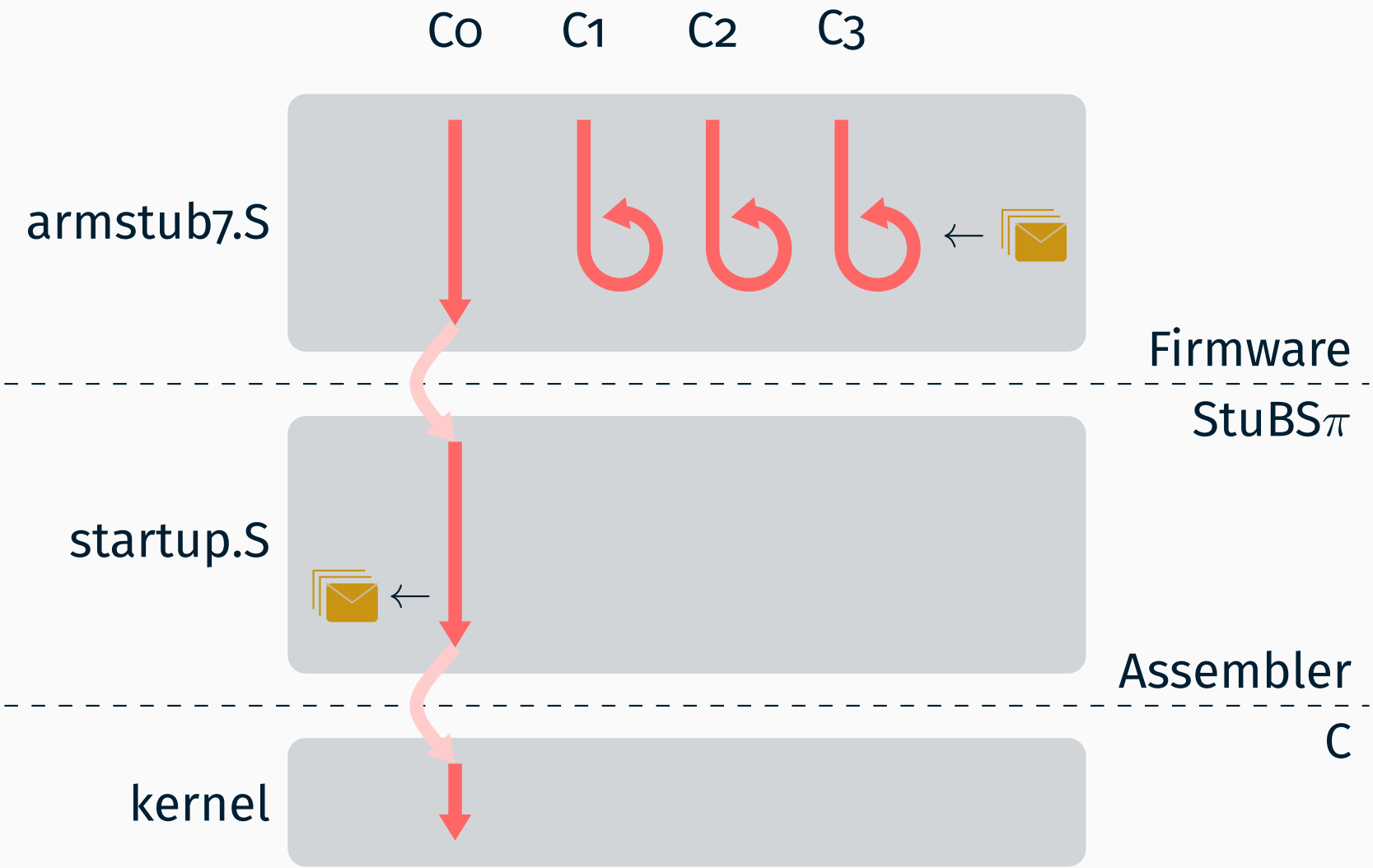
Raspberry Pi besitzt einen sehr “einzigartigen” Bootprozess.

- GPU: Feststellen der Bootmediums
- GPU: Laden von `bootcode.bin`
- GPU: Laden von `start.elf`
- GPU: Parsen von `config.txt`
- GPU: Laden des Kernelimages `kernel7/8.img`
- CPU: Ausführen von `armstub7/8.S`
- CPU: Ausführen des Kernels

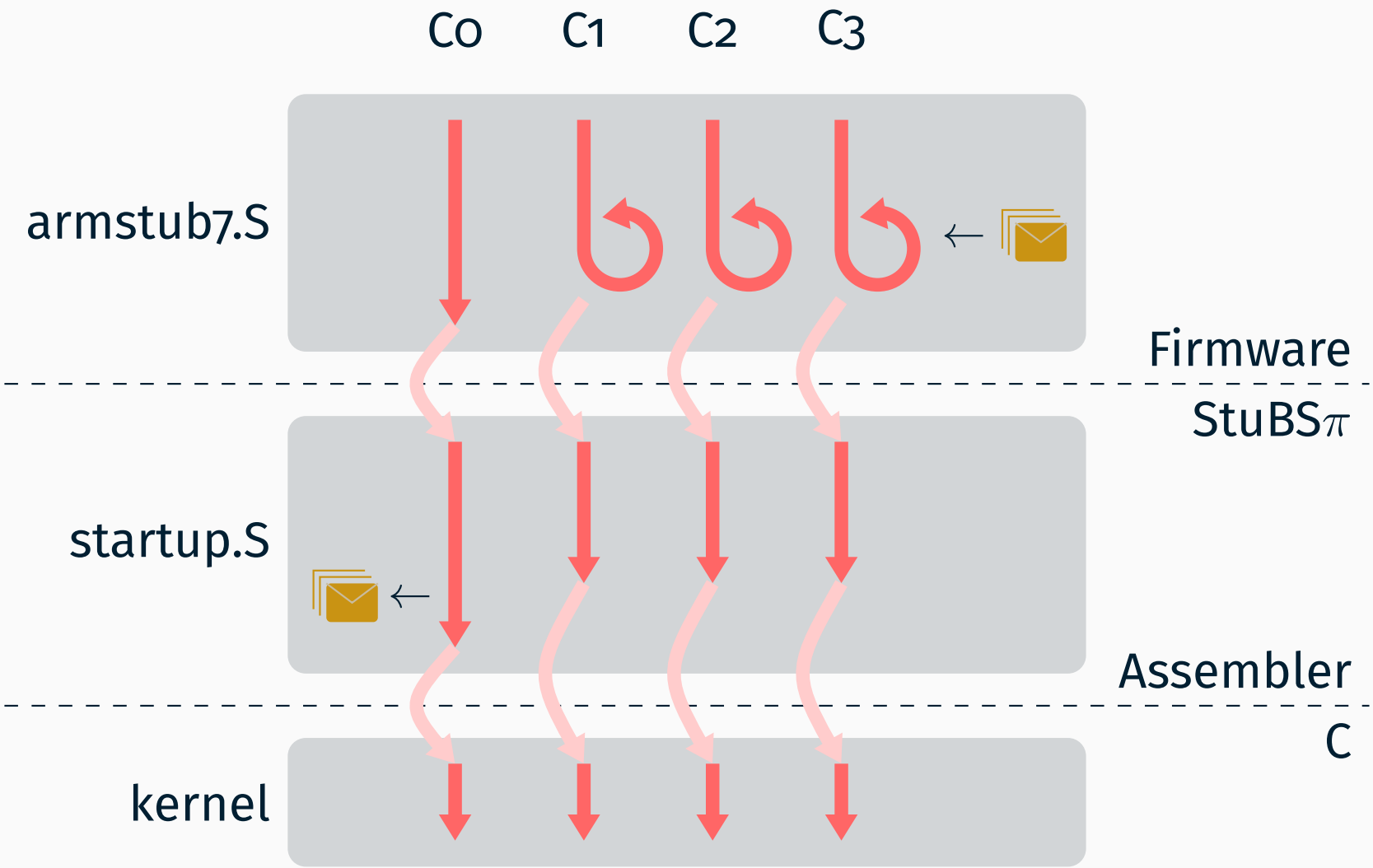
Startup


C0
C1
C2
C3
armstub7.S
Firmware
StuBSπ
startup.S
Assembler
C
kernel

Startup


C0
C1
C2
C3
armstub7.S
Firmware
StuBSπ
startup.S
Assembler
C
kernel

Startup


C0
C1
C2
C3
armstub7.S
startup.S
kernel
Firmware
StuBSπ
Assembler
C

armstub7.S

```
1:
    wfe                                 // wait for event
    ldr   r3, [r5, r7, lsl #4]          // load mailbox to r3
    cmp   r3, #0                        // compare r3 and 0
    beq   1b                            // go to 1 if r3 == 0
```

armstub7.S

```
1:
    wfe                                 // wait for event
    ldr   r3, [r5, r7, lsl #4]          // load mailbox to r3
    cmp   r3, #0                        // compare r3 and 0
    beq   1b                            // go to 1 if r3 == 0

    str   r3, [r5, r7, lsl #4]          // clear mailbox
    mov   r0, #0                        // load 0 to r0
    movw  r1, #machid                   // load machid to r1
    bx    r3                            // branch to r3
```

Mailboxes

- Jede CPU besitzt mehrere Mailboxes
- Dienen zum Nachrichtenaustausch zwischen CPUs
- Sind an feste Adressen gebunden (MMIO)

Mailboxes

- Jede CPU besitzt mehrere Mailboxes
- Dienen zum Nachrichtenaustausch zwischen CPUs
- Sind an feste Adressen gebunden (MMIO)
- Jeweils zwei verschiedene Adressen ($\rightarrow$ ermöglicht Synchronisation)
  - Register zum Schreiben
  - Register zum Lesen / Rücksetzen

Mailboxes

- Jede CPU besitzt mehrere Mailboxes
- Dienen zum Nachrichtenaustausch zwischen CPUs
- Sind an feste Adressen gebunden (MMIO)
- Jeweils zwei verschiedene Adressen ($\rightarrow$ ermöglicht Synchronisation)
  - Register zum Schreiben
  - Register zum Lesen / Rücksetzen
- Hier: **Übermittlung der Startadressen**

Globales Event

Kerne 1–3 warten passiv mit `wfe` (Wait For Event)

- Kern 0 muss Event signalisieren
- Befehl `dsb`: Data Synchronization Barrier
- Befehl `sev`: Set Event

Lösung

```
ldr r0, =_start             // load start addr. to r0
mov r1, #ARM7_MBOX_BASE // load mb base addr. to r1

// send start address to other cores
// store r0 in respective mailboxes
str r0, [r1, #ARM7_MBOX_OFF_C1_MB3]
str r0, [r1, #ARM7_MBOX_OFF_C2_MB3]
str r0, [r1, #ARM7_MBOX_OFF_C3_MB3]

dsb         // data synchronization barrier
sev         // set event -> wake up other cores
```

Portierung auf weitere Raspberry Pi Versionen

Version	Architektur	Erfolg	Notizen
RPi 2B	Cortex-A7	✓	Caches deaktivieren
RPi 2B v1.2	Cortex-A53	✓	Caches aktivieren
RPi 3B/3B+	Cortex-A53	✓	Caches aktivieren Bluetooth deaktivieren
RPi 4B	Cortex-A72	✗	Keine Verbindung mit HW-Debugger Keine Ausgabe über UART

Ausblick: 64-bit Portierung

- Einrichten eines 64-bit Cross-Compilers
- Portierung auf neues Instruction-Set
- Anpassen des Exception-Handling
- Anpassen der MMU-Konfiguration

Fragen?