

# **Dokumentation des OS**

eine Arbeit von

**Nicolaj Höss, Marko Petrović, Kevin Wallis**

**Master Informatik (ITM2)**

für die Lehrveranstaltung

**S1: Softwarelösungen für ressourcenbeschränkte  
Systeme**

Fachhochschule Vorarlberg

8. Mai 2015, Dornbirn

## **Abstract**

## Inhaltsverzeichnis

<b>1</b>	<b>Allgemein</b>	<b>4</b>
1.1	Aufbau . . . . .	4
<b>2</b>	<b>Architektur</b>	<b>5</b>
2.1	Aufbau . . . . .	5
<b>3</b>	<b>HAL</b>	<b>6</b>
3.1	Aufbau . . . . .	6
<b>4</b>	<b>Treiber</b>	<b>7</b>
4.1	Aufbau . . . . .	7
<b>5</b>	<b>Virtual Memory Management</b>	<b>8</b>
5.1	Aufteilung des virtuellen Speichers und Mapping . . . . .	8
5.2	Umwandlung virtueller Adressen zu physikalische Adressen . . . . .	11
5.3	Allokierung der Page Frames . . . . .	12
<b>6</b>	<b>Zusammenfassung</b>	<b>14</b>
6.1	xxx . . . . .	14
Literaturverzeichnis		15

## Abbildungsverzeichnis

1	Memory Map des Betriebssystems . . . . .	10
2	Umwandlung einer virtuellen Adresse durch die ARM CPU [1, S. B3-1337] . . . .	12
3	Beispiel einer Bitmap zur Verwaltung der Page Frames . . . . .	13

# **1 Allgemein**

bla

## **1.1 Aufbau**

bla

## 2 **Architektur**

bla

### 2.1 **Aufbau**

bla

3 *HAL*

## 3 HAL

bla

### 3.1 Aufbau

bla

## **4 Treiber**

bla

### **4.1 Aufbau**

bla

## 5 Virtual Memory Management

Bei der virtuellen Speicherverwaltung erfolgt die Umwandlung von vom ARM Prozessor generierten, virtuellen Adressen in physikalische Adressen durch die *Memory Management Unit* (MMU). Dieses Kapitel enthält die Beschreibung des Designs und der Implementierung der virtuellen Speicherverwaltung des Betriebssystems sowie der Einstellungen der MMU.

### 5.1 Aufteilung des virtuellen Speichers und Mapping

Die *VSMAv7* definiert zwei unabhängige Formate für translation tables [1, S. B3-1318]:

- *Short-descriptor format*:
  - zweistufige Seitentabelle
  - 32-bit Deskriptoren (PTE)
  - 32-bit virtuelle Eingangsadresse
  - bis zu 40-bit große physikalische Ausgangsadresse
- *Long-descriptor format*:
  - dreistufige Seitentabelle
  - 64-bit Deskriptoren (PTE)
  - verwendet *Large Physical Address Extension* (LPAE)
  - bis zu 40-bit große virtuelle Eingangsadresse
  - bis zu 40-bit große physikalische Ausgangsadresse

Um die Anforderungen an das Betriebssystem zu erfüllen, reicht das zweistufige Seitentabellensystem vollkommen aus. Tabelle 1 fasst die wichtigsten gegebenen Eigenschaften unter Verwendung des Short-descriptor format zusammen.

Eigenschaft	Beschreibung
Virtueller Speicher	4 GB
Größe eines Page Table Entry (PTE)	4 Byte
Einträge L1 Page Table	4096
Einträge L2 Page Table	256
Speicherbedarf L1 Page Table	4 Byte * 4096 = 16kB
Speicherbedarf L2 Page Table	4 Byte * 256 = 1kB
Unterstützte Pagegrößen:	<i>small page</i> (4 kB), <i>large page</i> (64 kB)
Unterstützte Sectiongrößen:	<i>section</i> (1 MB), <i>supersection</i> (16 MB)

Tabelle 1: Eigenschaften der virtuellen Speicherverwaltung der ARMv7-Architektur



### Seitentabellen

Der verwendete ARM Prozessor verfügt über zwei Register (*Translation Table Base Register*, *TTBR0* und *TTBR1*), welche Startadressen von Seitentabellen enthalten können [1, S. B3-1320]. Diese Register übernehmen die folgende Funktion:

- **TTBR0:** Wird für prozessspezifische Adressen verwendet. Jeder Prozess enthält eine eigene L1-Seitentabelle. Bei einem Kontextwechsel erhält das TTBR0 eine Referenz auf L1-Seitentabelle des neuen Kontextes/Prozesses.
- **TTBR1:** Wird für das Betriebssystem selbst und für memory-mapped I/O verwendet. Diese ändern sich bei einem Kontextwechsel nicht.

Abbildung 1 zeigt die Speicherverwaltung des Betriebssystems. Die rechte Seite stellt das physikalische Speichermapping dar und wurde dem Datenblatt des ARM [2, S. 155] entnommen. Die linke Seite zeigt die Aufteilung des virtuellen Speichers.

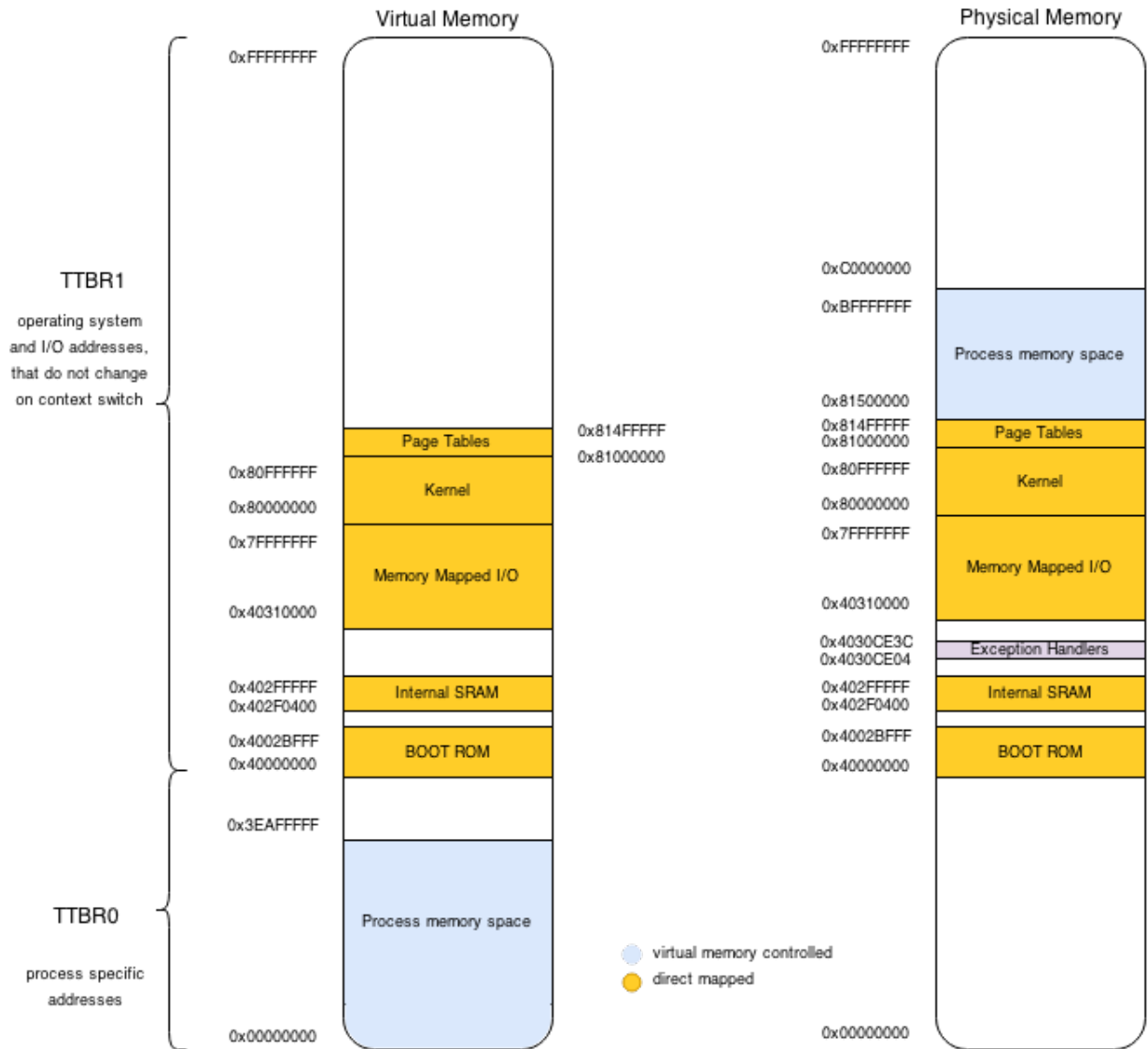


Abbildung 1: Memory Map des Betriebssystems

<b>Eigenschaft</b>	<b>Beschreibung</b>
Größe der Pages	4 kB
Speicherbedarf Kernel	16 MB
Virtueller Speicher für Prozesse	1003 MB
Physikalischer Speicher für Page Tables	5 MB
Max. Anzahl von L1 und L2 Page Tables	320 L1 Page Tables oder 1 L1 Page Table + 1276 L2 Page Table
Theoretisch Max. Anzahl von Prozessen	320

Tabelle 2: Eigenschaften der virtuellen Speicherverwaltung des OS

## 5.2 Umwandlung virtueller Adressen zu physikalische Adressen

Abbildung 2 zeigt die Umwandlung einer vom ARM Prozessor erzeugten virtuellen Adresse in eine physikalische Speicheradresse. Die Umwandlung wird vollständig durch die Prozessor-Hardware durchgeführt.

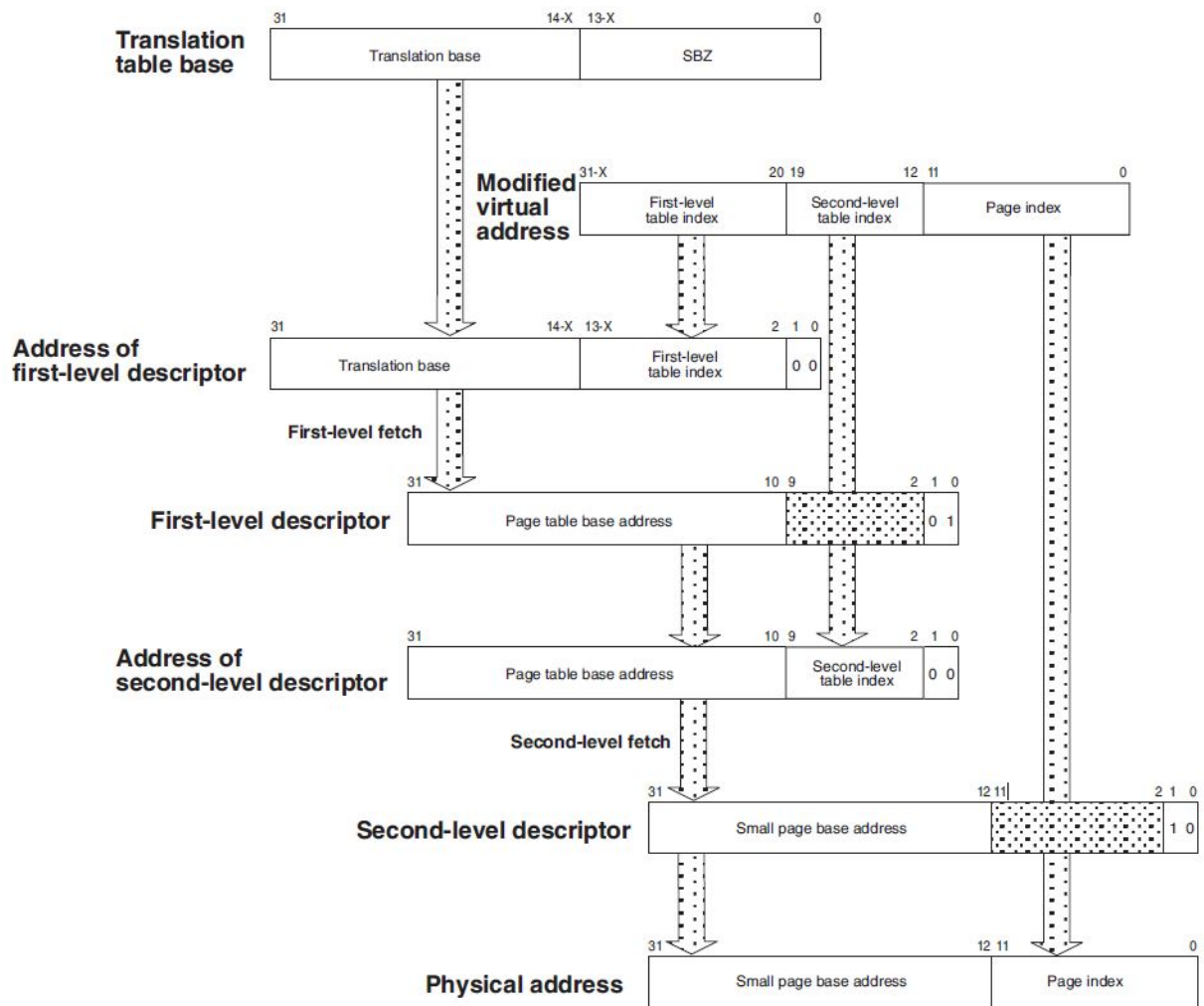


Abbildung 2: Umwandlung einer virtuellen Adresse durch die ARM CPU [1, S. B3-1337]

### 5.3 Allokierung der Page Frames

Für die Verwaltung der page frames wurde eine Bitmap verwendet. Abbildung 3 zeigt das Prinzip. Die Bitmap wird durch ein Array der Länge  $N/8$  Bytes realisiert.  $N$  steht hier für die Anzahl der page frames. Das  $i$ -te Bit im  $n$ -ten Byte der Bitmap definiert den Verwendungsstatus des  $(n*8 + i)$ -ten page frame.

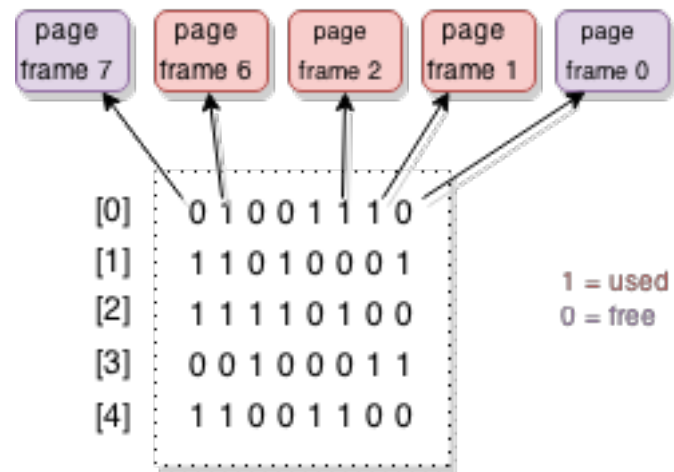


Abbildung 3: Beispiel einer Bitmap zur Verwaltung der Page Frames

## 6 Zusammenfassung

bla

[2]

[1]

### 6.1 xxx

bla

## Literatur

- [1] ARM Limited. *ARM Architecture Reference Manual ARMv7-A and ARMv7-R edition*, 2012. ARM DDI 0406C.b.
- [2] Texas Instruments. *AM335x ARM Cortex-A8 Microprocessors (MPUs) Technical Reference Manual*, 2011. Revised April 2013.