

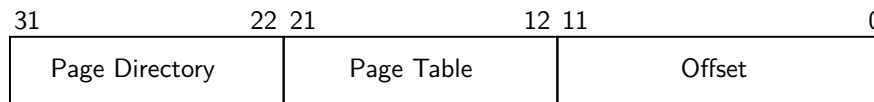
Betriebssysteme und Sicherheit, WS 2021/22

10. Aufgabenblatt – Virtueller Speicher I

Geplante Bearbeitungszeit: 20.12.2021 – 06.01.2022

Aufgabe 10.1 Im Folgenden betrachten wir die virtuelle Adressumsetzung auf der x86-Prozessor-Familie und gehen daher von folgenden Voraussetzungen aus:

- 32-Bit-Adressen
- Größe des virtuellen Adressraums: 4 GiByte
- Seitengröße: 4 KiByte
- zweistufige Seitentabellen: die ersten 10 Bit der Adresse dienen als Index für die erste Stufe (dem Seitenverzeichnis, Page Directory), die nächsten 10 Bit als Index für die zweite Stufe (den Seitentabellen, Page Table) und der Rest als Offset innerhalb der Seite:

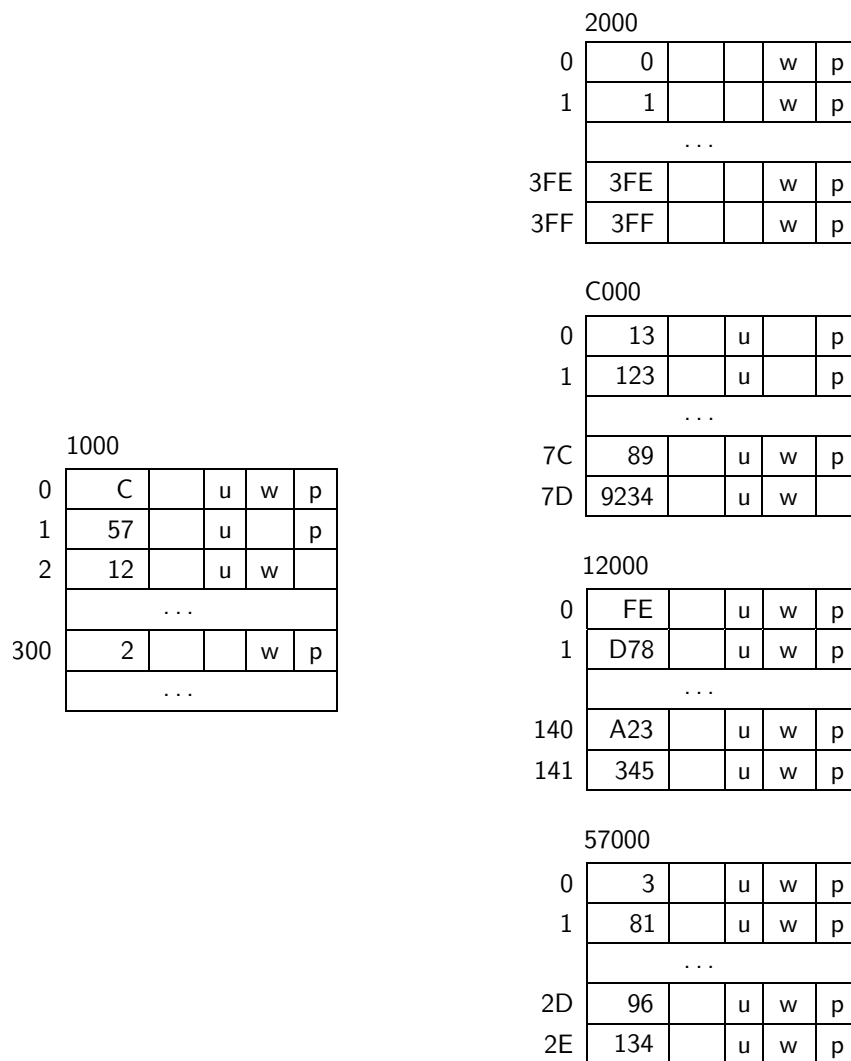


- ein Eintrag in einer Seitentabelle hat dabei folgende Bestandteile:
 - den Verweis auf die Kachel wo entweder die Seite hin abgebildet wird, oder wo die Seitentabelle der nächsten Stufe gespeichert ist
 - eine Reihe von Bits, die den Eintrag genauer beschreiben. Unter anderem folgende:

P	W	U	A	D
Present	Write	User	Accessed	Dirty

- (a) Diskutieren Sie die Bedeutung der jeweiligen Bits der Seitentableneinträge. Für was werden sie benötigt und wie werden sie interpretiert?
- (b) Gegeben sei nun der durch die abgebildete zweistufige Seitentabelle beschriebene Adressraum, wobei 0x1000 die Eintrittsadresse für das Page Directory ist. Alle angegebenen Zahlen sind Hexadezimal notiert, ein freies Feld in den letzten drei Spalten bedeutet, dass das entsprechende Bit nicht gesetzt ist. Betrachtet werde eine Menge von Speicherzugriffen aus dem normalen „user mode“ heraus (siehe Tabelle, Bedeutung der Abkürzungen siehe Aufgabe 8). Entscheiden Sie für jeden Speicherzugriff, ob der Zugriff gestattet ist und geben Sie in diesem Fall die physische Adresse an. Bestimmen Sie andernfalls den Grund für auftretende Seitenfehler und beschreiben Sie die Reaktion des Betriebssystems.

virt. Adresse	Zugriffsart	PD	PT	physische Adresse bzw. Grund für PF
0x1234	lesend			
0x42D8E4	lesend			
0x1256	schreibend			
0x400985	schreibend			
0x7DF87	schreibend			
0x940AFE	lesend			
0xC03FEB81	lesend			



Aufgabe 10.2 Im Unix-Teil der Vorlesung wurde die logische Struktur des Adressraums eines Unix-Prozesses vorgestellt. Die Bereiche innerhalb dieses Adressraums werden durch das Betriebssystem mittels geeigneter Datenstrukturen verwaltet.

- (a) Geben Sie einen möglichen Aufbau einer solchen Datenstruktur an, die die Informationen enthält, die zur Beschreibung der einzelnen Bereiche erforderlich sind.
- (b) Bilden Sie die folgende Situation mit Hilfe dieser Datenstruktur ab: Die Größe des Text-, Daten- und BSS-Segments eines Prozesses betrage 20 KiByte, 6 KiByte bzw. 11 KiByte (jeweils aufgerundet). Ferner habe der Prozess während seiner Laufzeit ab Adresse 0x5009000 einen 24 KiByte großen Ausschnitt aus einer Datei schreibbar eingeblendet, beginnend ab dem Datei-Offset 0x7000. Der Stack beansprucht aktuell eine Seite.
Hinweis: Bei einem 32-Bit System belegt das Betriebssystem im virtuellen Adressraum den obersten GiByte.

- (c) In diesem Adressraum sei auf folgende Adressen erfolgreich lesend zugegriffen worden und es gelte die nebenstehende Rahmenzuordnung. Schlagen Sie für die fehlenden Einträge geeignete Rahmen vor und konstruieren Sie die entsprechende einfache Seitentabelle.

virtuelle Adresse	physische Adresse
0x1000	0x340000
0x7000	0x32000
0x8000	
0xA000	
0x5009000	0x13000
0xBFFFF000	0x42000
0xC0000000	0
0xC0001000	0x1000

- (d) Was geschieht bei einem Lesezugriff auf die Adressen 0x3400, 0x500D100, 0xC0000040 und einem Schreibzugriff auf 0xA780, 0x3B060, 0xBFFFEDCB, wenn der Zugriff von einem normalen Nutzerprogramm aus erfolgt?

Aufgabe 10.3 Zum Erzeugen eines neuen Prozesses wird in Unix der Systemruf `fork()` benutzt. Der Adressraum des dadurch erzeugten Prozesses ist eine identische Kopie des Adressraums des erzeugenden Prozesses. Da in der Regel der neue Prozess sofort ein `execve()` ausführt und dabei den Inhalt seines Adressraums ersetzt, versucht man das tatsächliche Kopieren zu vermeiden. Die zugehörige Technik heißt *copy on write*. Erläutern Sie, wie *copy on write* prinzipiell funktioniert, und führen Sie es für den folgenden, durch eine einfache Seitentabelle beschriebenen Adressraum exemplarisch durch (die 2. Spalte der Tabelle werde dabei für das COW-Bit genutzt). Was geschieht, wenn nach dem Kopieren zunächst der Kind-Prozess und danach der Vater-Prozess schreibend auf Seite 7 zugreifen?

