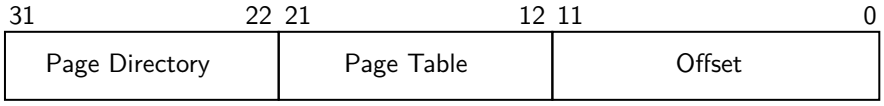
Betriebssysteme und Sicherheit, WS 2024/25

11. Aufgabenblatt – Virtueller Speicher I

Besprechungszeitraum: 21.01.2025 – 24.01.2025

Aufgabe 11.1 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	Acessed	Dirty

(a) Diskutieren Sie die Bedeutung der jeweiligen Bits der Seitentabelleneinträge. 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. 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			

1000

0	C		u	w	p
1	57		u		p
2	12		u	w	
	...				
300	2			w	p
	...				

2000

0	0			w	p
1	1			w	p
	...				
3FE	3FE			w	p
3FF	3FF			w	p

C000

0	13		u		p
1	123		u		p
	...				
7C	89		u	w	p
7D	9234		u	w	

12000

0	FE		u	w	p
1	D78		u	w	p
	...				
140	A23		u	w	p
141	345		u	w	p

57000

0	3		u	w	p
1	81		u	w	p
	...				
2D	96		u	w	p
2E	134		u	w	p

Aufgabe 11.2 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). Der Stack beansprucht aktuell eine Seite.
Hinweis: Bei einem 32-Bit System belegt das Betriebssystem im virtuellen Adressraum üblicherweise das oberste GiByte.

(c) Für den zuvor konstruierten Adressraum ist die folgende Seitentabelle gegeben (`x` = „execute“, `nx` = „no execute“, `u` = „user“, `s` = „superuser“, `ro` = „read-only“, rw = „read-write“):

	frame	execute	user	write	present
0					
1	340	x	u	ro	p
2					
3					
4					
5					
6					
7	32	nx	u	rw	p
8	36	nx	u	rw	p
9					
A	33	nx	u	rw	p
...					
BFFFF	42	nx	u	rw	p
C0000	0	?	s	?	p
C0001	1	?	s	?	p
...					

Was geschieht bei einem Lesezugriff auf die Adressen `0x3400`, `0xC0000040` und einem Schreibzugriff auf `0xA780`, `0x3B060`, `0xBFFFEDCB`, wenn der Zugriff von einem normalen Nutzerprogramm aus erfolgt? Ergänzen Sie die Seitentabelle, falls notwendig.

Aufgabe 11.3 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?