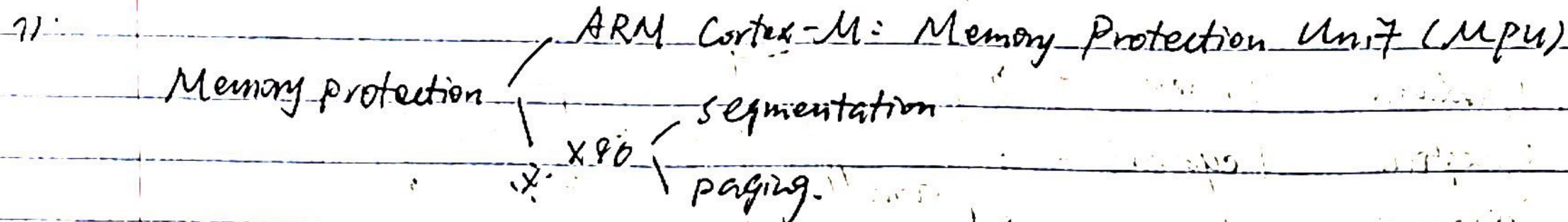


ch8. Intro to Paging



1.1) Segmentation

regs {
 fetching instrs: 'CS' (code segment)
 stack operations (pop/push): 'SS' (stack segment)
 other: 'DS' (data segment) or 'ES' (extra segment)
 additional: 'FS', 'GS'

- protected mode:

segment descriptors \rightarrow CDT/LDT

(offset + segment size + access permission)

- cons: fragmentation (external)

bootloader: sets the access permissions for each page

1.2) Paging

one-to-one
pages \rightarrow frames
(virt) (phy)

\rightarrow Our kernel runs on paging.
(内核中使用 virt addr)
(物理 VGA)

- cons: internal fragmentation (internal frag-)

virt 0xb8000 \rightarrow phy 0xb8000
(identity mapped)

- Page Table

p.s. page fault = CR2

{
 page + frame + flags.
 currently active table = 'CR3' (on x86)
 translation cache (page \rightarrow frame) = TLB

- Multilevel PT on x86

offset 1 2 3 4
4-level page table ([10=11], [12:20], [21:29], [30:38], [39=47])
a page of 4 KiB size (512 entries, each 8 bytes) id x
[48=63] = bit-47 (sign-extension). (As 5-level: [48=57])

- PTE (8 bytes)

(本条式见[8.3])
52 bits phy addr {
 [12:51]: physical address. (4096 bits aligned)
 \rightarrow [10=11] = 0
 flags
 available