

1.

Fair schedulers may equally partition the CPU among processes: for example, using Round-Robin scheduling which runs a job for a scheduling quantum and then switches to the next job. If the jobs take the same time, then average turnaround time would be increased.

Unfair schedulers such as Shortest Job First (SJF) or Shortest Time-to-Completion First (STCF) allow the shortest job to run first. Using SJF and STCF can decrease the turnaround time and increase the system throughput (number of processes completed per unit time) because shorter jobs are finished first.

2.

$$K = 2^{\{10\}}; M = 2^{\{20\}}; G = 2^{\{30\}}; T = 2^{\{40\}}; P = 2^{\{50\}}; E = 2^{\{60\}}$$

Number of virtual address bits	Number of virtual addresses	largest possible virtual address
4	$2^4 = 16$	$16 - 1 = 15$
8	$2^8 = 256$	$256 - 1 = 255$
16	$64K = (2^6)(2^{\{10\}}) = 65536 = 2^{\{16\}}$	$65536 - 1 = 65535 = 64K - 1$
32	$2^{\{32\}} = (2^2)(2^{\{30\}}) = 4G$	$4G - 1$
48	$256T = (2^8)(2^{\{40\}}) = 2^{\{48\}}$	$256T - 1$
64	$2^{\{64\}} = (2^4)(2^{\{60\}}) = 16E$	$16E - 1$

3

$$\begin{aligned} \text{page size} &= 4K = (2^2)(2^{\{10\}}) = 2^{\{12\}} \text{ bytes/page} \\ (2^{\{16\}} \text{ bytes}) / (2^{\{12\}} \text{ bytes/page}) &= 2^4 = 16 \text{ page table entries} \end{aligned}$$

$$\begin{aligned} \text{page size} &= 8K = (2^3)(2^{\{10\}}) = 2^{\{13\}} \text{ bytes/page} \\ (2^{\{16\}} \text{ bytes}) / (2^{\{13\}} \text{ bytes/page}) &= 2^3 = 8 \text{ page table entries} \end{aligned}$$

$$\begin{aligned} \text{page size} &= 4K = (2^2)(2^{\{10\}}) = 2^{\{12\}} \text{ bytes/page} \\ (2^{\{32\}} \text{ bytes}) / (2^{\{12\}} \text{ bytes/page}) &= 2^{\{20\}} = 1M \text{ page table entries} \end{aligned}$$

$$\begin{aligned} \text{page size} &= 8K = (2^3)(2^{\{10\}}) = 2^{\{13\}} \text{ bytes/page} \\ (2^{\{32\}} \text{ bytes}) / (2^{\{13\}} \text{ bytes/page}) &= 2^{\{19\}} = 512K \text{ page table entries} \end{aligned}$$

$$\begin{aligned} \text{page size} &= 4K = (2^2)(2^{\{10\}}) = 2^{\{12\}} \text{ bytes/page} \\ (2^{\{48\}} \text{ bytes}) / (2^{\{12\}} \text{ bytes/page}) &= 2^{\{36\}} = 64G \text{ page table entries} \end{aligned}$$

4.

32-bit virtual address space

24-bit physical address

VPN = virtual page number

PPN = physical page number

Offset tells which byte within the page

P	# of VPN bits	# of PPN bits	# of offset bits
1 KB = $2^{\{10\}}B$	$32 - 10 = 22$	(# of bytes of physical memory) / (page size) = $(2^{\{24\}}bytes)/(2^{\{10\}}bytes/page)$ = $2^{\{14\}}$ physical pages; so PPN is 14 bits	10
2 KB = $2^{\{11\}}B$	$32 - 11 = 21$	(# of bytes of physical memory) / (page size) = $(2^{\{24\}}bytes)/(2^{\{11\}}bytes/page)$ = $2^{\{13\}}$ physical pages; so PPN is 13 bits	11
4 KB = $2^{\{12\}}B$	$32 - 12 = 20$	(# of bytes of physical memory) / (page size) = $(2^{\{24\}}bytes)/(2^{\{12\}}bytes/page)$ = $2^{\{12\}}$ physical pages; so PPN is 12 bits	12
8 KB = $2^{\{13\}}B$	$32 - 13 = 19$	(# of bytes of physical memory) / (page size) = $(2^{\{24\}}bytes)/(2^{\{13\}}bytes/page)$ = $2^{\{11\}}$ physical pages; so PPN is 11 bits	13

## 5.1

On the context switch, the flush operation sets the valid bits to 0 and clearing the contents of the translation-lookaside buffer (TLB).

TLB misses happen when page frame number cannot be found inside the TLB.

5 TLB misses: 0x500, 2 times for 0x200000, 2 times for 0x300000 (on the context switch, the caches is emptied, so the accessing instruction 0x500 is the first access which would be a TLB miss because subsequent instructions 0x504 and 0x508 are likely to live in the same page as 0x500, so accessing 0x504 and 0x508 should be TLB hits)

(first time accessing 0x200000: TLB miss, translation is moved from disk to main memory)

(second time accessing 0x200000: TLB miss, translation is moved from main memory to TLB)

(third time accessing 0x200000: TLB hit, translation is found in the TLB)

(first time accessing 0x300000: TLB miss, translation is moved from disk to main memory)

(second time accessing 0x300000: TLB miss, translation is moved from main memory to TLB)

(third time accessing 0x300000: TLB hit, translation is found in the TLB)

## 5.2

Page faults are dynamic random access memory (DRAM) cache misses, and page faults occur when processes attempt to access pages on disk or if the pages do not exist.

2 page faults: 0x200000, 0x300000, (because these pages are on disk)