Solutions

Marking: **M1** a correct method, **A1** the correct answer, **B1** an independent mark.

Each answer shows the steps, then the **result**.

2.1

add two numbers: user mode; read a file: kernel

sort a list: user mode; send a packet: kernel

2.2

input (fd 0) from `in.txt`

normal output (fd 1) to `out.txt`

errors (fd 2) to `err.txt`

2.3

(a) **2**: parent and child both print

(b) **4**: each fork doubles the processes

2.4

1000×5 µs

$= 5$ ms a second

$= \mathbf{0.5}\%$

3.1

(a) **1**,**048**,**576**; 1,048,576/4096 = **256**

(b) $1{,}048{,}576 \times 0.5$ µs $\approx \mathbf{0.52}$ s; 256×0.5 µs = **128** µs

(c) each system call has a fixed cost; one large read replaces thousands of small ones, and later reads are served from the buffer in user memory

(d) e.g. more memory used, or a longer wait before the first data is ready

3.2

(a) **2**

(b) the child, where `fork` returned 0

(c) to pause until the child ends and collect its exit status; without it, `done` may print before `ls` finishes, and the finished child stays as a zombie until collected

(d) after `ls` has finished

3.3

(a) 1/20 ms = **50** per second

(b) 2/20 = **10**%

(c) CPU limit 1/2 ms = **500** per second; about 20/2 = **10** threads keep the CPU busy

(d) $2 \times 5 = 10$ µs; 10 µs/2 ms = **0.5**%

(e) no: the CPU is already full at 10 threads, so more threads only add switching and memory; more cores, or less CPU work per request, would help

exercises.lol exercises.com.cn