CS2106 Tutorial 11

AY 25/26 Sem 1 — github/omgeta

Q1.

a. ReadExeDir: Y — WriteExeDir: N — ExeOnlyDir: N

b. ReadExeDir: Y — WriteExeDir: Y — ExeOnlyDir: Y

c. ReadExeDir: Y — WriteExeDir: N — ExeOnlyDir: N

d. ReadExeDir: Y — WriteExeDir: Y — ExeOnlyDir: Y

e. ReadExeDir: Y — WriteExeDir: Y — ExeOnlyDir: Y

f. ReadExeDir: N — WriteExeDir: Y — ExeOnlyDir: N

Q2.

a. 12KB

b. "/y/i" -¿ 0, 3 (pass)
"/x/z/i" -¿ 0, 1, 2 (fail)

c. "/x/z/k" -¿ block 6 ":)"
"/y/h" -¿ block 27, 30, 21, 14, 4, 23, 7, 10, 28 "OPERATINGSYSTEM:-("

d. Bitmap: bits 5, 8, 13, 15, 16 set to 0 (ALLOCATED)
Directory: block 3 (for /y) added "n —File— 5—16"
Data: blocks 5, 8, 13, 15, 16 next block pointer changed with -1 in block 16

Q3.

a. Requests are of nearby sectors, and are of same type (e.g. read or write)

b. Seeking latency is reduced

c. Starvation possible if request is not near existing requests. Mitigation: add deadlines from start time and ensure that if deadline is reached then we must take the request

d. Disk IO has high latency so delays can be beneficial if we can wait for more requests for merging and efficient operations.

e. If they use different algorithms, the scheduling by the OS might be overwritten by the hardware controller cauisng the time used by the OS for sorting and merging to be wasted.