File Management Schemes (15 points)

Assume an indexed file management scheme that is single indirect. There are 10 direct pointers in the inode. The 11th and 12th entries in the inode are BOTH pointers to single indirect index blocks. An index block can hold 1000 pointers to physical blocks. Each physical block holds 4000 bytes.

a. (5 pts) What is the largest possible file that can be stored using this scheme? Show your work.

\[
(10 + 2 \times 1000) \times 4000 \text{ bytes} = 80,000,000 \text{ bytes}
\]

b. (10 pts) Suppose a file of size 200 blocks is stored in this system. Suppose the probability of accessing the first 10 blocks is 2% for each individual block and that the probability of accessing the last 10 blocks is also 2% for each individual block. The remaining blocks are all equally likely. What is the effective access time to access a block in this file. Assume time to access a direct block is \((m+d)\) and time to access a single indirect block is \((2m + 2d)\).

\[
p(1-10) = 10 \times 2\% = 20\% \\
p(111-200) = 20\% \\
\text{Therefore } p(11-190) = 60\%
\]

\[
\text{EAT} = \left(0.2)(m+d) + (0.2)(2m+2d) + (0.6)(2m+2d)\right) \\
= 1.8(m+d)
\]

-3 if they have \(\text{EAT} = (0.2)(m+d) + (0.2)(2m+2d) + (0.2)(2m)\) because they thought the whole range had 02 probability.

-6 if wrong prob. for 11-190

-6 if extra sum with 20%, 50%, 20%
**Effective Access Time (10 points)**

Assume a double indirect indexed file organization. The inode has 10 direct pointers and each index block holds 100 pointers. What is the effective access time to access a block in a file of size 100 blocks if the first 20 blocks are twice as likely to be accessed as the last 80 blocks? Use the fact that the time to access a direct block is \((m + d)\) and the time to access a single indirect block is \((2m + 2d)\). You won't need to access any double indirect blocks.

\[
p = \text{prob. of accessing one of the last 80 blocks} \\
p = \frac{20}{200} \\
p = \frac{1}{10}
\]

\[
EAT = 10 \cdot 2p (m + d) + 10 \cdot 2p (2m + 2d) + 80 \cdot p \cdot (2m + 2d)
\]

\[
= \frac{1}{10} (m + d) + \frac{2}{10} (2m + 2d) + \frac{2}{10} (2m + 2d) = \frac{11}{10} (m + d)
\]

because first 10 blocks are direct and remaining 90 are single indirect.

**Design of Indexed File System (10 points)**

Suppose you buy a hard drive that has a total of \(2^{32}\) physical blocks, each of size \(2^{12}\) bytes. You are designing an indexed file organization for this disk.

a. How many addresses can fit in a physical block? (Hint: first compute the number of bits per block address based on the number of physical blocks.)

\[
\text{Need 32 bits to address } 2^{32} \text{ blocks}
\]

\[
\frac{2^{12} \cdot 2^3 \text{ bits/block}}{32 \text{ bits/addresses}} = 2^{10} \text{ addresses/block}
\]

b. How many levels of indirection are needed if the biggest single file uses all \(2^{32}\) blocks? Assume that the inode contains 10 direct pointers. You know how many pointers per block from (a). (Assume you don't need any blocks on disk for the inode or any free list information.)

\[
\text{Need } 10 + 2^{10} + (2^{10})^2 + (2^{10})^3 + ... (2^{10})^n > 2^{32}
\]

\[
10 + 2^{10} + 2^{12} + 2^{15} + 2^{20} > 2^{32}
\]

\[
\Rightarrow \text{quad indirect file system}
\]