Final Review

Section A01& A02

Borun Song, Hanran Wu, Stuti Shah

Announcements

- Exam 4 will be held 2:50pm on July 28th in lecture hall
- Extra Credit Project will be due on July 31st at 11:59pm
- Homework 11 (ungraded) has been released. Use it as a practice

CIOS Incentive

If we are able to get 85% response rate, there will be 1% added towards the final score.

Piazza Participation

- There is a 5% participation grade in your final grade, and 2% is coming from piazza participation.

- “Piazza participation accounts for 2% (Provide answers to peer questions; Ask questions; Work out past exams collaboratively, etc.); Note: We will use the summary stats from Piazza in the following categories: "views", "contributions", "questions", "answers". The exact weightage of each is not revealed to the students, but I am sure the students are savvy enough to know which would count for more!”

Overview of Review Session Today

- IO
- Disk
- File System
- Networking

I/O

Disk Drive

Unix File System Command

Unix command	Semantics	Elaboration
touch <name>	Create a file with the name <name>	Creates a zero byte file with the name <name> and a creation time equal to the current wall clock time
mkdir <sub-dir>	Create a sub-directory <sub-dir>	The user must have write privilege to the current working directory (if <sub-dir> is a relative name) to be able to successfully execute this command
rm <name>	Remove (or delete) the file named <name>	Only the owner of the file (and/or superuser) can delete a file
rmdir <sub-dir>	Remove (or delete) the sub-directory named <sub-dir>	Only the owner of the <sub-dir> (and/or the superuse) can remove the named sub-directory
ln –s <orig> <new>	Create a name <new> and make it symbolically equivalent to the file <orig>	This is name equivalence only; so if the file <orig> is deleted, the storage associated with <orig> is reclaimed, and hence <new> will be a dangling reference to a non-existent file
ln <orig> <new>	Create a name <new> and make it physically equivalent to the file <orig>	Even if the file <orig> is deleted, the physical file remains accessible via the name <new>
chmod <rights> <name>	Change the access rights for the file <name> as specified in the mask <rights>	Only the owner of the file (and/or the superuser) can change the access rights
chown <user> <name>	Change the owner of the file <name> to be <user>	Only superuser can change the ownership of a file
chgrp <group> <name>	Change the group associated with the file <name> to be <group>	Only the owner of the file (and/or the superuser) can change the group associated with a file
cp <orig> <new>	Create a new file <new> that is a copy of the file <orig>	The copy is created in the same directory if <new> is a file name; if <new> is a directory name, then a copy with the same name <orig> is created in the directory <new>
mv <orig> <new>	Renames the file <orig> with the name <new>	Renaming happens in the same directory if <new> is a file name; if <new> is a directory name, then the file <orig> is moved into the directory <new> preserving its name <orig>
cat/more/less <name>	View the file contents	

Q2 Unix File Systems

24 Points

A friend is learning how to use the file system on her Linux distribution and has some questions about how it works. She runs `ls -l` and gets the following output on her current directory:

```
-rw-r--r-- 2 sam  2200tas 4591 Nov 22 23:10 backup-ta-info.txt
-rwxr-xr-- 1 sam  2200tas 1809 May  2 21:23 calc-scores.py
drwxr-xr-x 1 sam  2200tas    0 Nov 22 22:41 exam-answers/
drwxr-xr-x 1 aj   2200tas    0 Nov 23 02:31 homework-answers/
lrwxrwxrwx 1 jack 2200tas   33 Aug 27 23:35 projects -> ~/prj/
-rw-r--r-- 2 sam  2200tas 4591 Nov 22 23:10 ta-info.txt
```

Problem

Given the following specifications for a disk drive:

256 bytes per sector
12 sectors per track
20 tracks per surface
3 platters
Average seek time of 20 ms
Rotational speed 3600 RPM

- What would be the time to read 6 contiguous sectors from the same track?

- What would be the time to read 6 sectors at random?

Solution

- Time to read 1 random sector
  = Avg seek time + Avg rotational latency + Sector read time
  = a + (60 / (r * 2)) + (60 / (r * s)) seconds
  = 0.02 + (60 / (3600 * 2)) + (60 / (3600 * 12)) = 0.0297 seconds
- Time to read 6 contiguous sectors
  = 0.02 + (60 / (3600 * 2)) + ((60 / (3600 * 12)) * 6) = 0.0366 seconds
- Time to read 6 random sectors
  = (0.02 + (60 / (3600 * 2)) + (60 / (3600 * 12))) * 6 = 0.1783 seconds

File System

Problem

Given the following:

Size of index block = 512 bytes
Size of data block = 2048 bytes
Size of pointer = 8 bytes (to index or data blocks)

The i-node consists of

2 direct data block pointers,
1 single indirect pointer, and
1 double indirect pointer.

An index block is used for the i-node as well as for the index blocks that store pointers to other index blocks and data blocks. Note that the index blocks and data blocks are allocated on a need basis.

(a) What is the maximum size (in bytes) of a file that can be stored in this file system?
(b) How many data blocks are needed for storing the same data file of 266 KB?
(c) How many index blocks are needed for storing a data file of size 266 KB?

Solution

Size of index block = 512 bytes
Size of data block = 2048 bytes
Size of pointer = 8 bytes (to index or data blocks)

The i-node consists of
2 direct data block pointers,
1 single indirect pointer, and
1 double indirect pointer.

a) Maximum file size:

An index block can hold 512/8 = 64 entries

2 direct data block pointers in i-node

64 entries in first-level index

64 * 64 entries in second-level index

Total: 4162 blocks * 2048 bytes = 8,523,776 bytes

b) Number of data blocks to hold 266KB:

266 * 210 / 2048 = 266 / 2 = 133 blocks

(c) How many index blocks to hold 266KB (133 blocks):

2 direct + 64 first-level blocks + (64 + 3) second-level blocks

1 i-node + 1 first-level index + 1 + 2 second-level index

Networking

Problem

What is the bit stream represented by the following Manchester encoding of the stream?

Problem

Given the following:

Message size	=	1900 Kbits
Header size per packet	=	1000 bits
Packet size	=	20 Kbits
Bandwidth on the wire	=	400,000 bits/sec
Time of flight	=	2 secs
Window size	=	10
Processing Delay at the Sender	=	0
Processing Delay at the Receiver	=	0
Size of ACK message	=	negligible (take it as 0)

Assuming an error free network and in-order delivery of packets, what is the total time to accomplish the above message delivery?