CS 241 Section Week #12
(04/30/09)

Announcements

- TA Final review:
  - -Either Tuesday May 12, 2009
  - -Or Wednesday May 13, 2009
  - (2:00pm - 4:00pm) || (6:30pm - 8:30pm) || (7:00pm - 9:00pm)
  - (we will announce it on the newsgroup about time and location)
- **No Discussion Section next week**

Outline

- SMP5
  - Goal and Task
  - HTTP
  - Code Recap
- HW2
- Extra Questions

SMP5

SMP5

Goal: Implement a simple content-filtering web proxy

- Check the blacklist for blocked hostname
- Add specific advertisement according to the domain type

SMP5 Tasks

1. Set up TCP sockets with the client (web browser such as Firefox)
2. Parse the request from the client to get the hostname, check whether that host is blacklisted
3. In case the host is not blacklisted, forward the request to the original server and then get the response back from the server
4. Before forwarding the response to the client, look at the content type of the response
   - if the content is of type "text/html", look at the domain type (org, com, net, edu, etc.) to put customized advertisement along with the original response from the server

HyperText Transfer Protocol

HTTP: Parsing the URL

Each *Uniform Resource Locator* (URL) has three components:

http://www.linux.org/info/faq1.html

Protocol Host Path

HTTP

Hypertext Transfer Protocol

- Delivers virtually all files and resources on the World Wide Web
- Uses Client-Server Model

HTTP transaction

- HTTP client opens a connection and sends a request to HTTP server
- HTTP server returns a response message

Sample HTTP exchange

Scenario

- Client wants to retrieve the file at the following URL (http://www.somehost.com/path/file.html)

What a client does

- Client opens a socket to the host www.somehost.com, port 80
- Client sends the following message through the socket

```
GET /path/file.html HTTP/1.0
From: someuser@uiuc.edu
Host: www.somehost.com
User-Agent: HTTPTool/1.0
[blank line here]
```

Sample HTTP exchange

What a server does

- Server responds through the same socket

```
HTTP/1.0 200 OK
Date: Mon, 17 Apr 2006 23:59:59 GMT
Content-Type: text/html
Content-Length: 1354

<html>
<body>
(more file contents)
  .
  .
  .
</body>
</html>
```

HTTP: Request and Response

Request or response message Format:

1. A request or response line
2. Zero or more header lines
   - Request: Host
   - Response: content-type
3. A blank line (CRLF)
4. An optional message body.

Networking Basics

Generic TCP Client & Server Code Recap

Client	Server
socket()	socket()
connect()	bind()
	listen()
while (…) {	while (…) {
	accept()
send()/recv()	send()/recv()
}	}
close()	close()

Homework #2

Q1: A server is 95% utilized. Assuming Poisson arrivals and exponentially distributed service times, if the average service rate is 10 requests/second, what is the average queuing time for an incoming request?

a) 0.1 sec b) 0.95 sec c) 1.8 sec d) 1.9 sec e) 2 sec

Q1: A server is 95% utilized. Assuming Poisson arrivals and exponentially distributed service times, if the average service rate is 10 requests/second, what is the average queuing time for an incoming request?

a) 0.1 sec b) 0.95 sec c) 1.8 sec d) 1.9 sec e) 2 sec

Q2: If the incoming request rate on the above server dropped until its average utilization became 90%, what is the average number of requests in the entire system?

a) 0.9 b) 8.1 c) 9 d) 10

b) e) None of the above

Q2: If the incoming request rate on the above server dropped until its average utilization became 90%, what is the average number of requests in the entire system?

a) 0.9 b) 8.1 c) 9 d) 10

b) e) None of the above

Q3: Please research disk scheduling policies and familiarize yourself with shortest seek time first (SSTF). *Hint: SSTF is a disk scheduling policy that attempts to serve the closest track to the current track next.* A disk head starts at track 60, when requests for the following tracks arrive simultaneously: 50, 58, 25, 70, 110. Which track is served by SSTF last?

a) 25 b) 50 c) 58 d) 70 e) 110

Q3: Please research disk scheduling policies and familiarize yourself with shortest seek time first (SSTF). *Hint: SSTF is a disk scheduling policy that attempts to serve the closest track to the current track next.* A disk head starts at track 60, when requests for the following tracks arrive simultaneously: 50, 58, 25, 70, 110. Which track is served by SSTF last?

a) 25 b) 50 c) 58 d) 70 e) 110

Q4: A file system uses 4-byte pointers to refer to locations of file blocks on disk. Each block is 1024 bytes. If the file index is not allowed to exceed 10 blocks in total, what approximately is the maximum allowable file size? (Below, 1KB = 1024bytes and 1MB = (1024)2bytes)

a) 10 KB b) 256 KB c) 2.5 MB d) 40 MB e) None of the above

Q4: A file system uses 4-byte pointers to refer to locations of file blocks on disk. Each block is 1024 bytes. If the file index is not allowed to exceed 10 blocks in total, what approximately is the maximum allowable file size? (Below, 1KB = 1024bytes and 1MB = (1024)2bytes)

a) 10 KB b) 256 KB c) 2.5 MB d) 40 MB e) None of the above

Q5: A 32-bit machine partitions its memory into 1024 byte pages. Approximately how long is the page table if it was loaded entirely in memory? Assume each entry in the table takes six bytes.

a) 1 Mbytes b) 4 Mbytes c) 6 Mbytes d) 24 Mbytes c) None of the above

Q5: A 32-bit machine partitions its memory into 1024 byte pages. Approximately how long is the page table if it was loaded entirely in memory? Assume each entry in the table takes six bytes.

a) 1 Mbytes b) 4 Mbytes c) 6 Mbytes d) 24 Mbytes c) None of the above

Q6: If the sizes of the largest available hole were compared across the following contiguous memory allocation policies, which policy will likely have the largest hole more often?

a) Best fit b) Worst first c) First fit d) Next fit e) Random fit

Q6: If the sizes of the largest available hole were compared across the following contiguous memory allocation policies, which policy will likely have the largest hole more often?

a) Best fit b) Worst first c) First fit d) Next fit e) Random fit

Q7: You are designing a NetFlix video server to serve movies from the "Oldies" category to clients via the Internet (you may assume that the set of movies in that category does not change). The server is optimized for sequential video streaming. Which of the following methods would you choose for movie file allocation?

a) Contiguous b) Indexed c) Linked d) Linked Indexed e) Multilevel Indexed

Q7: You are designing a NetFlix video server to serve movies from the "Oldies" category to clients via the Internet (you may assume that the set of movies in that category does not change). The server is optimized for sequential video streaming. Which of the following methods would you choose for movie file allocation?

a) Contiguous b) Indexed c) Linked d) Linked Indexed e) Multilevel Indexed

Q8: In some machine architecture, the TLB access time is 0.1 msec. The memory access time is 1 msec. The TLB hit ratio is 0.4. You can assume that the page table needs a single memory access. What is the effective memory access tine?

Q8: In some machine architecture, the TLB access time is 0.1 msec. The memory access time is 1 msec. The TLB hit ratio is 0.4. You can assume that the page table needs a single memory access. What is the effective memory access tine?

Answer: Effective access time = 1.1 * 0.4 + 2.1 * 0.6 = 1.7 microseconds.

Q9: What is the effective memory access in the above machine, if the page table needed two accesses on average?

Q9: What is the effective memory access in the above machine, if the page table needed two accesses on average?

Answer: Effective access time = 1.1 * 0.4 + 3.1 * 0.6 = 2.3 micrseconds

Q10: Which of the following machines is most likely to use an inverted page table? (lengths are of virtual addresses)

a) 8-bit machine b) 16-bit machine c) 32-bit machine d) 64-bit machine e) machine with no virtual memory

Q10: Which of the following machines is most likely to use an inverted page table? (lengths are of virtual addresses)

a) 8-bit machine b) 16-bit machine c) 32-bit machine d) 64-bit machine e) machine with no virtual memory

Extra Questions to Practise

Question #1 (I/O Devices)

QUESTION #2 (I/O DEVICES) [25 points]

A. Suppose a machine has a hard drive with the following characteristics:

- Rotational speed (rpm): 20,000
- Bytes per sector: 256
- Sectors per track: 800
- Tracks on the disk: 2000
- Average seek time (ms): 10

a) What is the expected/average rotational delay for this disk? ***(2 points)***

Answer: The average rotational delay is defined as half of the amount of time it takes for the disk to make one complete rotation.

Therefore: (1 minute / 20,000 rotations) * 0.5 =
0.000025 minutes =
0.0015 seconds =
1.5 ms

b) What is the total time to read one sector on this disk? ***(2 points)***

Answer: One sector on the disk is $1/800^{th}$ of a track. The physical time it would take is $1/800^{th}$ the time of one rotation of the disk.

Therefore: (1 minute / 20,000 rotations) * (1 / 800) =
6.25×10^{-6} minutes =
3.75×10^{-4} seconds =
0.00375 ms

However, the total time it takes to read one sector requires the disk to both seek to the correct sector and rotate to the correct position on the disk. The total time to read a single sector is then:

[Seek Time] + [Rotational Delay] + [Physical Read Time] =
10 ms + 1.5 ms + 0.00375 ms =
11.50375 ms

QUESTION #2 (I/O DEVICES) [25 points]

A. Suppose a machine has a hard drive with the following characteristics:

- Rotational speed (rpm): 20,000
- Bytes per sector: 256
- Sectors per track: 800
- Tracks on the disk: 2000
- Average seek time (ms): 10

c) What is the total time to read one track on this disk? ***(2 points)***

Answer: To read an entire track on a disk, the physical read operation requires an entire rotation of the disk. From (a), we know that the time of half a rotation of a disk is 1.5 ms. Therefore, a full rotation would require 3ms.

The total time to read an entire sector would then be:

[Seek Time] + [Rotational Delay] + [Physical Read Time] =
10 ms + 1.5 ms + 3.0 ms =
14.5 ms

However, modern hard drives have the capability of caching information before we reach the beginning of the sector – allowing us to read immediately once we've have seeked to the correct track. Therefore, with basic caching, the total time to read an entire sector would be:

[Seek Time] + [Physical Read Time] =
10 ms + 3.0 ms =
13.0 ms

QUESTION #2 (I/O DEVICES) [25 points]

A. Suppose a machine has a hard drive with the following characteristics:

- Rotational speed (rpm): 20,000
- Bytes per sector: 256
- Sectors per track: 800
- Tracks on the disk: 2000
- Average seek time (ms): 10

d) What is the transfer time for a 10KB file that is stored contiguously on a single platter? ***(2 points)***

Answer: To find the total transfer time of a 10 KB file, we need to find the total amount of sectors that 10 KB is spread across.

10 KB file * (1024 B / 1 KB) * (1 sector / 256 B) =
40 sectors

Following the logic done in part (b):

(1 minute / 20,000 rotations) * (40 / 800) =
0.15 ms

And calculating the total transfer time:

[Seek Time] + [Rotational Delay] + [Physical Read Time] =
10 ms + 1.5 ms + 0.15 ms =
11.65 ms

QUESTION #2 (I/O DEVICES) [25 points]

A. Suppose a machine has a hard drive with the following characteristics:

- Rotational speed (rpm): 20,000
- Bytes per sector: 256
- Sectors per track: 800
- Tracks on the disk: 2000
- Average seek time (ms): 10

e) What is the average time to access a random byte inside the file mentioned above? ***(2 points)***

Answer: A random byte in the file would require reading a random amount of the file. Given an infinite number of reads, the average time would be equivalent to reading the center byte of the file.

From (d), we found that the physical time reading the file takes 0.15 ms. Reading half of that file would take 0.075 ms. Since all the other factors are fixed:

[Seek Time] + [Rotational Delay] + [Physical Read Time] =
10 ms + 1.5 ms + 0.075 ms =
11.575 ms or 11.5ms

B. Answer the following questions about *programmed I/O, interrupt-driven I/O* and *Direct Memory Access (DMA)*:

a) What is cycle stealing? ***(2 points)***
Answer: Cycle stealing is when a DMA operations steals a single CPU cycle to perform a DMA operation instead of consuming the entire system bus.

b) Which of the above three I/O mechanisms does it apply to? ***(2 points)***
Answer: Direct Memory Access (DMA)

c) Describe three conditions under which programmed I/O is preferable to interrupt-driven I/O. (*HINT: think about the characteristics of a process or of the overall system*) ***(3 points)***
Answer: A number of answers may be given here; ensure the answer is reasonable and different than the other answers.

Question #2 (File Systems)

QUESTION 4 (FILE SYSTEMS) [25 points]

A. The starting of a free space bitmap looks like the following after the disk partition is first formatted:

1000 0000 0000 0000 (the first block is used by the root directory).

The system always searches for free blocks starting at the lowest numbered block. Show the bitmap after each of the following actions:

a) File A with 6 blocks is written ***(1 point)***

Answer: 1111 1110 0000 00000

b) File B with 4 blocks is written ***(1 point)***

Answer: 1111 1111 1110 00000

c) File A is deleted ***(1 point)***

Answer: 1000 0001 1110 00000

d) File C with 7 blocks is written ***(1 point)***

Answer: 1111 1111 1111 00000

e) File B is deleted ***(1 point)***

Answer: 1111 1110 0001 00000

B. Suppose a UNIX file system has 4 KB blocks and 8B disk addresses. What is the maximum file size if i-nodes contain 10 direct entries, and one single, double, and tripe indirect entry each? (Remember: each file has only one i-node that points to other data blocks or index tables) ***(8 point)***

Answer: We can construct the total size of the files by looking at each of the i-node pointers.

Direct Pointers: 10 pointers * 4 KB/block = 40 KB

Single Indirect:
4 KB page filled with 8 B disk addresses
8 B / 4 KB = 512 entries pointing to disk blocks
512 entries * 4 KB/block = 2 MB

Double Indirect:
512 entries pointing to single indirect blocks
512 single indirect blocks * 2 MB /single indirect = 1 GB

Triple Indirect:
512 entries pointing to double indirect blocks
512 * 1 GB /double indirect = 0.5 TB

Total Size:
40 KB + 2 MB + 1 GB + 0.5 TB =
0.500 978 507 TB =
513,001 991 GB =
525,314,039 MB =
537,921,576 KB =
550,831,693,824 bytes

C. Let us consider the following shell commands on a local i-node based file system.

cd /tmp	[No i-nodes; no files.]
mkdir subdir	[i-node for subdir; subdir directory file]
echo "hello" > subdir/file1	[i-node for s/file1; s/file1 file]
ln -s subdir/file1 file2	[i-node for file2; file2 symlink file]
ln -s subdir/file1 file3	[i-node for file3; file3 symlink file]
ln subdir/file1 file4	[Updated file1 i-node]
ln file4 file5	[Updated file1 i-node]
chmod 744 file4	[Updated file1 i-node]
touch file3	[Updated file3 i-node]
rm file5	[Updated file1 i-node]
chmod 000 subdir	[Updated subdir i-node]

a) How many disk blocks were allocated in total? ***(2 points)***
Answer: A total of four i-nodes were created and four small files were created. Therefore, 8 disk blocks were used.

b) How many disk blocks were freed in total? ***(2 points)***
Answer: No disk blocks were freed.

c) Which command(s) incremented or decremented an existing i-node reference count? ***(2 points)***
Answer: Three commands incremented or decremented an existing i-node's reference count. Those commands were:
ln subdir/file1 file4
ln file4 file5
rm file5

C. Let us consider the following shell commands on a local i-node based file system.

cd /tmp	[No i-nodes; no files.]
mkdir subdir	[i-node for subdir; subdir directory file]
echo "hello" > subdir/file1	[i-node for s/file1; s/file1 file]
ln -s subdir/file1 file2	[i-node for file2; file2 symlink file]
ln -s subdir/file1 file3	[i-node for file3; file3 symlink file]
ln subdir/file1 file4	[Updated file1 i-node]
ln file4 file5	[Updated file1 i-node]
chmod 744 file4	[Updated file1 i-node]
touch file3	[Updated file3 i-node]
rm file5	[Updated file1 i-node]
chmod 000 subdir	[Updated subdir i-node]

d) Describe in words the permissions on the file '*file4*' ***(2 points)***

Answer: The permissions on file4 are set such that the owner of the file may read, write, and execute the file while the users in the group that the file is contained in and all other users may only read from the file and cannot write or execute the file.

e) Identify the last command to modify the inode of '*file3*' and describe which i-node field is modified. ***(2 points)***

Answer: The command 'touch file3' modified the i-node of file3 by updating its last access time.

f) Describe the indirect effect (if any) of the '*chmod 000 .*' command on the files *file1,file2,file3,file4*. ***(2 points)***

Answer: By running 'chmod 000' on the directory, no user may view the contents of the directory. Since file1, file2, file3, and file4 all access a file contained inside subdir, none of those files will be able to access their content since the contents of the directory may not be viewed by any user.