Final exam review

Final format
   11 questions
   10 points each
   short answer.

What's definitely on the final
I/O and filesystems
literacy: explain why
knowledge boundaries
scheduling
Basic principles of taking my exams:
I tend to be rough on irrelevant answers.
  At best I ignore them.
  At worst I deduct for it.
=> You should insure that you're answering the question that I asked.
=> Read and revise your answer after you write it.
  Make sure it answers the question.
  Make sure it covers all possible interpretations.
For short essay answers:
  Make an outline first.
  Critically assess the outline and whether it's complete.
  THEN write the paragraph.
  (use the liberal white space on the back for outlines. If you run out of time, I WILL give partial credit for an outline!)

To approach a problem:
  Determine what's relevant. Where in the course did the problem come from?
  Determine what's different.
  Make a decision on technique and implement.

Warnings:
  There may be several correct answers (and an infinity
of incorrect ones!
There may be a "new twist" or not.
Twists may be obvious or extremely subtle.
How I grade:
  pick out and mark the relevant points.
  pick out and mark the incorrect statements.
  pile exams into equivalence classes.
  each equivalence class gets the same grade.

To get to the top of the pile
  Nail the relevant points.
  Avoid irrelevant/incorrect statements.
  Don't spend time on potentially irrelevant things.
Comments on survival
Wednesday, December 15, 2010  6:52 PM

Survival

Answer the questions you're sure about, first.
Avoid peripheral issues.
After an initial pass, "grade" each of your answers:
  Did you cover the key points?
  Is anything unsure?
Surviving a short-answer question

Get to the point fast!
Try not to write down peripherally relevant things
Unless you have extra time...
Many review sheet problems are "long answer"
  Open-ended situations.
  Often a complex answer.
  Lots of flexibility.
Many final exam problems are "short answer"
  More limited situations.
  Simpler answer.
  Less degrees of freedom.
I claim that the disk pager is a producer-consumer architecture. What is the producer and what is the consumer?

```
sstrwxrwxrwx
ugouuuugggooo
u = user
g = group
o = other (everyone else)
750 is an octal code:
000 = 0
001 = 1
010 = 2
011 = 3
...
111 = 7
750 = 111 101 000
        rwx rwx rwx
```

```
chmod 700 file === chmod u+rwx go-rwx
```

<table>
<thead>
<tr>
<th>protection</th>
<th>files</th>
<th>directories</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>can read</td>
<td>can ls</td>
</tr>
<tr>
<td>w</td>
<td>can write</td>
<td>can create or delete files</td>
</tr>
<tr>
<td>x</td>
<td>can execute as</td>
<td>can open a file</td>
</tr>
</tbody>
</table>
user 1:
mkdir foo
chmod 0777 foo

user 2:
cd foo
mkdir bar
cd bar
touch nothing

user 1: can't get rid of bar, because it's non-empty and user 1 can't delete "nothing".

Q: Your web server says your public_html is unreadable but you check, and it's 755. What else can be a problem?

A: Your home directory is 0700 by default. You need to change it to 0705???. Nope, you just need 0701.

Q: Why do you need only 001?
A: Other entities than "group other" can open files.
   Apache runs as web:web, not "group other"
   Apache can open contents.
Q: what is the danger of this?
A: can access files that are readable and for which you know the name

What is sst:

<table>
<thead>
<tr>
<th>code</th>
<th>files</th>
<th>directories</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>set-user-id: run as file owner</td>
<td>meaningless</td>
</tr>
<tr>
<td>s</td>
<td>set-group-id: run as file group</td>
<td>group inheritance</td>
</tr>
<tr>
<td>t</td>
<td>meaningless</td>
<td>override normal file creation protections: only user can remove files the user created.</td>
</tr>
</tbody>
</table>

example:
cp /bin/csh foo
chmod u+sx go+rx foo
Then anyone can run ./foo, and they become me.

What is memory addressing? It is not on the final!
This is a long issue.

Several subtopics:
process and kernel descriptors
What a descriptor means
reason for descriptors
scope of descriptors.
Notes from extra session on 2015-12-14

Little's law
- 1/rate = interval (inter-arrival time)
- In \( L = \lambda W \),
  - \( \lambda \) is an average rate
  - 1/\( \lambda \) is inter-arrival average time
  - \( W \) = time in system
  - 1/\( W \) = service rate.
  - \( L \) = jobs in system.

A typist types 100 chars/minute, and a character is processed in 1/10 second; how many characters are in system at a time?

\[
\lambda = \frac{100}{\text{minute}} \times \frac{1}{60} \text{ seconds} = \frac{10}{6}
\]
\[
W = \frac{1}{10}
\]
\[
L = \lambda W = \frac{10}{6} \times \frac{1}{10} = \frac{1}{6}.
\]

A web request arrives every second and it takes 1/2 second to respond. What is the number of web requests in system?
\[
\frac{1}{2} = W
\]
\[
\lambda = 1
\]
\[
L = \lambda W = \frac{1}{2}
\]

The load average is 3 and there are 5
requests/second. What is the time of service?
$L = 3$
lambda = 5
$L = \lambda W$
$3 = 5 W$
$W = 3/5$ seconds

We have a Producer/Consumer system with two stages.
The first Producer reads bytes from a music stream at 100/second.
The first consumer turns this into midi at a rate of 200/second.
This is the second producer.
The second consumer sends this to the audio device at 150 midi commands per second.
What is the delay between first and last stages?

\[
\frac{1}{\lambda_1} = 100 \\
\frac{1}{\mu_1} = 200 \\
\frac{1}{\lambda_2} = 150 \\
\]

\[
W = \frac{1}{\mu_1} = \frac{1}{200} \\
W = \frac{1}{\lambda_2} = \frac{1}{150} \\
\text{delay} = \frac{1}{\mu_1} + \frac{1}{\lambda_2} \\
\]

$x = \text{jobs/time}$
\[ W = \text{time} \]
\[ L = \text{jobs} \]
\[ \min(m, m_2) \]
For M/M/1 queue,
Queue reaches steady state only if \( p = \lambda / \mu < 1 \)
Prob(queue length is \( k \)) = \( p^k (1-p) \) where \( p = \lambda / \mu \)
The mean jobs in system are \( n = p / (1-p) \)
(including jobs in queue and jobs being processed).
The mean waiting-line length (jobs not yet being served) is \( w = p^2 / (1-p) \) (including jobs in process of being queued and dequeued).

A web server receives requests at a rate of 50 per minute and processes them one at a time. It takes 1/2 second to process one in an exponential distribution. What is the delay between request and response?

It's M/M/1, lambda = 50/60 = 5/6 per second.
mu = 1/(1/2) = 2 per second.
lambda / mu = (5/6)/2 = 5/12 = p
jobs in system = p/(1-p) = 5/12 / 7/12 = 5/7.
time in system = 5/7 / lambda = 5/7 / 5/6 = 6/7

\[ n = \frac{p}{1-p} \]
\[ w = \frac{n}{\lambda} \]
Where should a journal be on a disk?