Read carefully and follow the rules below.

1. This is an open book, and open notes exam. Computers can only be used after disabling the wireless network.

2. This exam consists of 10 pages and 9 problems. Take the space provided as a hint of the length of the answers expected. Do not add additional pages to the exam.

3. Make sure that you explain clearly each step in your reasoning; no credit is given if an answer is not justified.

4. Each student is expected to work independently; acts of academic dishonesty lead to an unconditional failure of the class.

5. Write legibly and do not use a pencil; if I cannot read an answer you do not get the points.

6. You have 75 minutes for the exam.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>100</strong></td>
</tr>
</tbody>
</table>
Problem 1 (10 points: 1 + 3 + 6)

The page ranking algorithm is at the heart of one of the most used application of the client-server paradigm.

1.1 Name the application and use two words to describe what the algorithm models.

1.2 Give the formula for the page ranking algorithm and explain each entity involved.

1.3 Provide a qualitative justification for each term in this formula.

*Hint:* for example, if $PageRanking(A) = a + b + c$ justify why $PageRanking(A)$ can be modelled as the cumulative effect of three factors, $a, b,$ and $c$; if the term $b = d/e$ justify why $b$ is directly proportional with $d$ and inverse proportional with $e$. 
Problem 2 (10 points: 7 + 3)

You have a system with \( n \) components and you decide to organize it as a collection of \( s \) subsystems.

2.1 Give a practical example illustrating the advantages of such a strategy.

2.2 Can you relate the expression \( \log_s n \) with an aspect of this organization?
Problem 3 (10 points: 5 + 5)

3.1 The original version of the Unix system had a number of innovations; name more than two and justify why each one was innovative in your view. It also lacked a number of features that became part of the UNIX system in later years: name more than two and justify the need for each.

3.2 The Unix file system is organized in several layers; some of the layers in fact implement a virtualization scheme. Name the layers and explain the role of each one in this virtualization scheme. Can you associate this scheme with the flavors of virtualization discussed in class?
Problem 4 (10 points: 5 + 5)

4.1 Remote procedure call (RPC) implementations take also advantage of virtualization concepts to hide the details of communication. Justify this statement and explain how RPC implements virtualization?

4.2 What is “fate sharing” in the context of procedure calls? Do RPCs prevent fate sharing?
Problem 5 (10 points)

The Harvard architecture physically separates storage and signal pathways for the instructions and the data. The term originated from the Harvard Mark I computer built at Harvard University. Soft modularity of systems based on von Neumann architecture has several weaknesses. Does the Harvard architecture reduce the possibility of errors due to soft modularity? If yes, discuss the specific errors that can be prevented by the Harvard architecture.
Problem 6 (20 points: 10 + 3 + 7)

6.1 A system allows passwords consisting of combinations of $n$ 8-bit characters selected from a set of 128 characters. The system has the following features:
(i) it reports a reference to an unassigned virtual page by an interrupt to the user program;
(ii) a system call is viewed as a machine instruction for an extended machine, and any reference it makes to an unassigned virtual page is thus similarly reported to the user program;
(iii) to obtain access to another directory, a user invokes a system call implemented by the following code

```plaintext
for i = 0 to Length[DirectoryPassword] do
    if DirectoryPassword[i].ne>PasswordArgument[i] then
        Wait three seconds;
        return BadPassword;
    endloop
Connect to directory;
return Success;
```

One of the arguments of this system call is a string containing the password for the directory. If the password is wrong, the call fails after a three second delay, to prevent guessing passwords at high speed. A bug allows to guess a password with $n$ characters in $64n$ tries on the average, rather than $\frac{256^n}{2}$. Describe step-by-step the procedure to exploit this bug.

6.2 Justify the expressions $64n$ and $\frac{256^n}{2}$.

6.3 This is an example when generality could lead to unexpected complexity. Explain the generality aspect(s) that led the designers of this system to allow this bug to creep in.
Problem 7 (10 points: 5 + 5)
A network file system (NFS) uses a file handle to access remote files.

7.1 Justify the need for the information provided by each element included in the file handle.

7.2 What strategies are used by NFS to optimize performance.
Problem 8 (10 points: 1 + 5 + 4)

The publish-subscribe paradigm is widely used nowadays.

8.1 Give three practical examples?

8.2 Consider an event service in a distributed system. List all the primitive functions an event service based on this paradigm must implement; explain how each of these functions should be used.

8.3 Does this paradigm support “thin clients?” If the answer is yes, explain why this solution is better than an alternative one, you could think of; else justify why not.
Problem 9 (10 points)

A protocol is a communication discipline involving an exchange of a finite sequence of messages. A communication channel has a probability $p$ of losing any message. Can you design a protocol allowing two entities to reach agreement. If the answer is yes then sketch the protocol; else give a rigorous justification of why not.