Quiz 1

Due Oct 12 at 11:59pm **Points** 50 **Questions** 5 **Available** Oct 12 at 12am - Oct 12 at 11:59pm about 24 hours **Time Limit** 30 Minutes

Instructions

You have to work on the quiz independently. **You cannot discuss with any other people.** Any form of discussion is considered cheating.

This is open book and open note quiz. But **no search on the internet**. You can access the reading list and the paper pdf files listed in the reading list on Canvas.

This quiz was locked Oct 12 at 11:59pm.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	29 minutes	50 out of 50

ⓘ Correct answers are hidden.

Score for this quiz: 50 out of 50
Submitted Oct 12 at 6:07pm
This attempt took 29 minutes.

Question 1

10 / 10 pts

Paper: Plan 9 From Bell Labs
Question: Plan 9 proposed to have dedicated machines as file servers or CPU servers, and the user-side machine can be just a terminal. Name two benefits of such design.

Your Answer:

1. User terminal won't need to have much power -- it can even don't have storage section as the server can handle them

2. Permanent Record: it has WORM as backup, where it is hard for just home machine to keep these data.

1. The user-side machines are be simple and even disk-less

2. When the file servers or CPU servers upgrade, all users can benefit.

3. The users don't need to worry about maintaining and managing file servers and CPU servers. Those servers can have dedicated admins for maintenance tasks such as backup and upgrades.

Question 2

10 / 10 pts

Paper: The Structure of the 'THE'-Multiprogramming System
Question: Name *two* benefits and *two* disadvantages of the structure/layered design.

Your Answer:

Advantages:

1. Easier to debug

2. Prove of correctness

Disadvantage:

1. Large overhead/inefficient

2. Less flexibility (fix hierarchy)

Benefits: (1) Easy to debug; (2) easy to verify

Disadvantages: (1) Less efficient as it has too many layers; (2) Hard to design since it cannot well support circular dependencies.

Question 3

10 / 10 pts

Paper: Protection and the Control of Information Sharing in Multics
Paper: The Unix Time-Sharing System

Question: Multics claims that it can check every access to every object efficiently for protection. How did it accomplish this? Can Unix do this too? What are the trade-offs in terms of protection approach.

Your Answer:

Multics requires hardware support for this check protection, and Unix cannot do so. The trade-off here is obvious -- Unix does not require such hardware to run, where Multics, even though provided higher level of security but the hardware limited its popularity. Unix's portability V.S. Multics strict protection

Multics relies hardware checks on the capabilities stored in each segment descriptor at run time. So each access to each object is checked for access right. Since it is done via hardware, it has good efficiency.

The downside for this of this approach include

---Difficult to port to other hardware that does not provide such hardware support

--- Inflexible since all accesses to files need to go through memory by first mapping into virtual memory first

Unix doesn't rely on hardware support for access checks. It is more portable and flexible. Access to files are via system calls and access rights are checked by the file system code, aka software.

Question 4

10 / 10 pts

Paper: Plan 9 From Bell Labs

Question: What proposed in Plan 9 are similar to today's cloud systems such as Google Drive, Gmail, Amazon AWS. Why?

Your Answer:

First, from the user terminal perspective, it can easily outplay the power of current PC, lower the performance/power requirement for individuals. Real world example: Windows 365. This probably because users are moving toward mobile working environment. The energy assumption and the size of the device are important

Also, the function definitions of WORM are basically what most cloud drive service provider must have -- except the non-deletion part maybe. The reliability and permanent record proposed in Plan 9 are also the trend today, especially for some industry e.g. gov, bank/finance.

Local namespace is still in use for the isolation, security, and bring possible for multi-tenant, which max the resource cost-effectiveness. Companies nowadays may borrow resources on demand and lower the cost of buying expensive machines but leave it idle when not in use, and each of them will have their own space without interfering each other's workload. Cloud computing such as AWS could help lowering companies' the cost because of this.

Dedicated File servers ---> Google Drive, Dropbox, etc.

Dedicated CPU servers ---> Cloud machines such as Amazon AWS, Microsoft Azure, Google Cloud

Question 5

10 / 10 pts

Paper: Pilot: An Operating System for a Personal Computer
Question: Why do systems today still use time-sharing systems? Is the security assumption made in Pilot still valid today, and why or why not?

Your Answer:

It is mostly like because that even though we won't share our person computers often, we still need to have some level of scheduling between the applications due to various user demanding e.g. multi-tasking and responsiveness. The security assumptions made in Pilot were not valid due to the Internet access. It is possible that hackers could run malicious software without direct physical access to the machine, or even user may need to run some unknown source software on their computer where isolation and security are demanded.

Although today's personal computers typically have only one user, this user usually have many processes running such as browsers, emails, powerpoint, etc. So we still need time-sharing systems to allow these processes share computer resources (CPU, memory, network, etc).

The security assumption made in Pilot is not valid due to Internet. Today almost every computer is connected on the network, and malicious users may attack machines from any part of the world. In 1980s, systems such as Pilot didn't need to worry much about attackers on the Internet.

Quiz Score: 50 out of 50