Distributed Operating System

- Multiple machines, no shared memory

- Looks like one single OS (or integrated system)

- Old as networking, software, and tools for cooperation
While I "sense"...

- not everyone knows, companies, and

- not everyone can read/update

- keep schedule/Calendars

- users (Synchronization)

Calendar Application
- Limited concurrency

- 2 per transaction

- num of users?

- easy to manage

- all logic on server

- smooth, fast

- simple

- secure, load

Can be extended.
Distributed → no server

At least 2 methods

1) - every 'client' machine stores its owners data (partitioned)

2) - all machines store all data (replicated)
(2 weeks) \( \geq 2(n-1) \) weeks

- weak and aggregate

- leaves & edges: synchronous

- many normal functions

- coder \( \rightarrow \) local (0 message)

- ask updates & help

All \( \rightarrow \) approach #1
Approach #2

- Alice reads Alice -> local
  - Bob's key
  - everyone
  - and key
  - Bob is

→ Alice updates... → "Alice"

"Alice reads Alice" -> local, "Alice"
Multiple OS (or apps) that look like 1 OS or app?
→ Calendar app: yes.
Left parahippocampal gyrus
Left entorhinal cortex
Left hippocampus
Right entorhinal cortex
Right hippocampus
Left fusiform gyrus
Dist OS model
- Asynchronous server-side rendering (DAN)

  - Remote service

  - Death Life System Care to

    - Remote Procedure Call (RPC)

    - Message Passing

    - More examples
- Scarcity
  - In Good
  - 4 mandatory of DIP & replacement
  - Reliability
  - Efficient
  - Argument
  - Compress

Preparation of destruction
Last time do not ask service.

done circuit switching

setup

calendar on

done

check

keep and x
Muddy leaves.
After leaves
of neighbors (first neighbors)
- Inexact - (vs speech)
- Grammatically = work vs dot
Other issues