Plan 9 from Bell Labs

## motivation

trend: networks of smaller computers

the UNIX: centralized time-sharing system,

A distributed system composed of different PCs which do different tasks, such as small machines as terminals and central machines as server

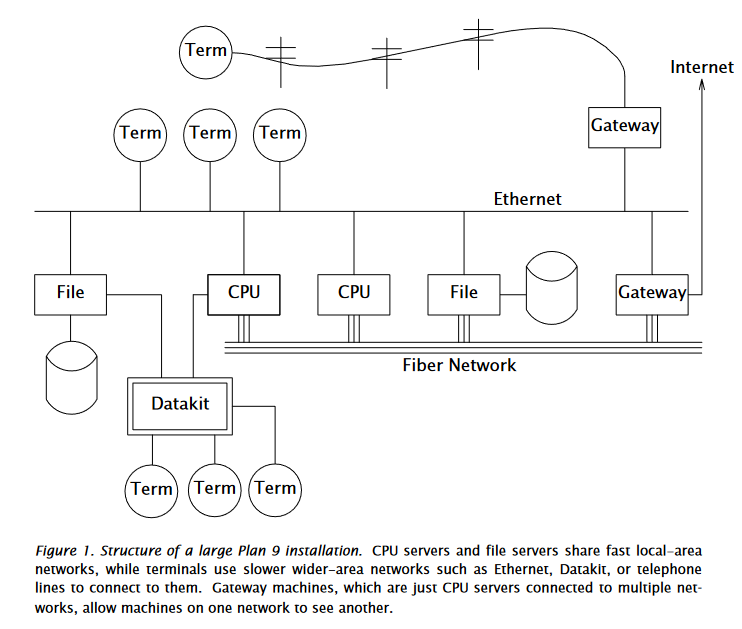
Design:

network-level protocol 9P to enable machines to access files on remote systems;

naming system to build customized views of the resources in the network

=> adopt a relatively brand new design

## design



each terminal has a personalized name space for the public resources in the global network.

9P protocol: send a request from the local client to the server and return the result.

\*\*file-oriented\*\*.

## the command-level view

window system: has its own name space; created running an application;



run interactive applications on the terminal and run the computation or file-intensive applications on remote servers.

## the file server

centralized file server:

100MB memory buffer, 27G magnetic disks, 350GB WORM

WORM: daily backup(dump)

* user dump
* entire system dump(like a disaster happens)

WORM files are not able to be changed or removed once written.

## unusual file servers

file-like interface to unusual service - usually user-level processes

e.g. 8 ½(window system) - for user, windows who has applications running on; for client programs, a set of files like mouse, screen and cons.

8 ½ offers multiplex sets of private files to different clients

it can toggle the request, like, you can type multiple lines in the same time; and, the client read the text on the window in a real-time manner, allowing the user to change the next command line before the client really executes that line.

ftpfs: a 9P protocol of ftp

exportfs: make a portion of the own name space available for other processes.

* import: server content -> local
* cpu: local content -> server

## configurability and administration

Everthing is in central server, no private files, no decentralized administration

make the server best, the terminal simplest.

## C programming

ANSI C -> a dialect of C which have some extensions.

ANSI C requires prototype for all functions

#if is what?

no dynamic parameters for a single function

introduce Unicode and UTF-8

## portability and compilation

portable across a variety of processor architectures.

* avoid byte order problem by encoding data into texts.
* present data through a file system interface e.g. the process states are all stored in /proc file

the loader of the architecture is selective, but the compiler to generate in selective intermediate can be installed on any kind of architectrue. In PLAN9, there are different kinds of compilors and loaders who have unique names.

PLAN9 is across-plaform

## parallel programming

parallel: 1)kernel parallel; 2) the programming language is parallel

processes && threads

system call : rfork, with parameters different to create different kinds of resource allocation and sharing.

share memory: 1) by setting the flag of rfork 2) by using segattach system call to make the segment shared between the parent and the child.

synchronization: rendezvous system call

spin clock: architecture-dependent library at user-level

## implementation of name spaces

name space construction: mount, bind, unmount

union directory: the concatenation of the constituent directories, which means that differen directories are treated as a single directory

by default, no new files will be created under the union. Except that some directories ine union has the flag of allowing creation.

kernel file system: /dev

all kinds of operations are based on 9P protocol, and file pointer to the file in the remote server are called fids.

9P protocal: 17 messages

procedure of constructing name space: 1) message session 2) message attach

each fid is associated with a channel,

(这里没看懂、、、啊啊啊啊啊啊、、、、、、、救命、、、、)

## file caching

local file caching server: check the file version in the local cache,

## networks and communication devices

network information is also stored as file in the kernel, any setup or shutdown should be achieved by editting the file.

setting up a connection:

* write a connect message with a network-specific address
* talk to the network which is established by writing and reading the file \*data\*.

## kernel structure for networks

channel is composed of \*streams\*, which is a double-end vector.

stream: a bidirectional channel connecting a physical or pseudo-device to a user process.

## The IL protocol

IL(internet link) to transmit 9P messages over IP, provides reliable transmission of sequenced messages between machines.

IL has adaptive timeouts.

## overview of authentication

client: the user requesting the resource

server: the user granting access to the resource

user -> DES authentication key

bilateral

authentication server: maintain the database for the keys

speak-for relationship: CPU servers run process on behalf of client

## authenticating external connections

For text-based services such as Telnet or FTP, the user uses a one-time 4-digit pin.

## special users

* admin
* none: no password

## the cpu command and proxied authentication

* listener forks off a process to handle the call from the user
* the CPU performs authentication
* The CPU reaches out to the file server by speaking for the user.

## file permissions

user

user group

chmod: change the rights

the files in the dump should maintain the file permissions -> the user name is hard to re-use.