The Design and Implementation of Plan B 3rd edition.

A Plan 9 descendant for pervasive computing.

Francisco J. Ballesteros

Laboratorio de Sistemas
http://lsub.org/who/nemo

Roadmap

- Antecedents: Plan 9 & Plan B 2nd ed.
- Computing environments
- Paradigm shift
- System architecture
- Resource volumes
- Dynamic mounts
- System interface

Antecedents

Plan 9

- Everything (almost) is a file
- Network file system protocol (9P)
- Per-process customizable name spaces

Antecedents

Plan B 2nd ed.

- Hosted system
- Everything is a box
- No file descriptors
- Per-process prefix table

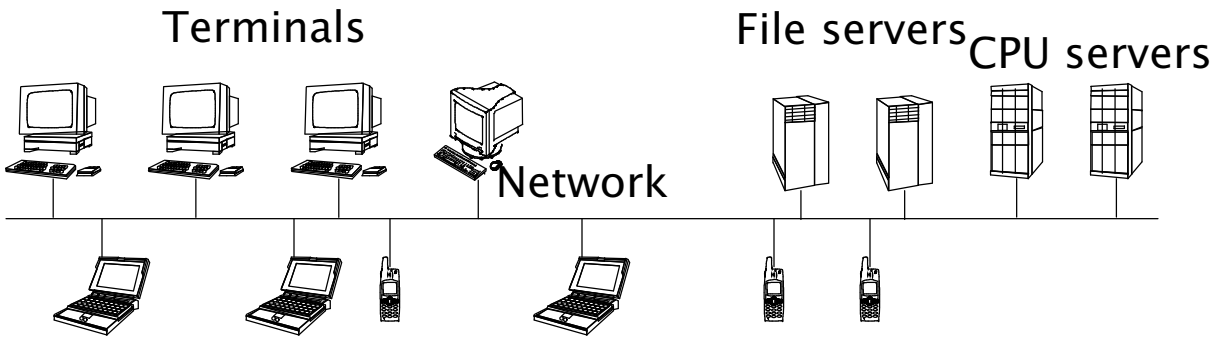
Antecedents

What happen?

- 2nd ed. kernel was complete
- We wanted to use it asap
- Enough services were already in Plan 9
- Idea:
  - Put Plan B into Plan 9.
  - But keep the interface as it is.

Computing environments

What changed since Plan 9?



What do we want?

Dynamism + Heterogenity + Services == Pervasive

What do we have?

Heterogeneity + Serv-

What do we have to do?

-ices + Dynamism

Paradigm shift: Files as boxes

Boxes:

- No descriptors
- constraints
- list op.

Files:

- Descriptors
- No constraints
- Files vs directories

Files as boxes

What can we do?

- Don’t keep files open
- Add constraints to mounted volumes
- Must keep directories

Files as boxes

New calls

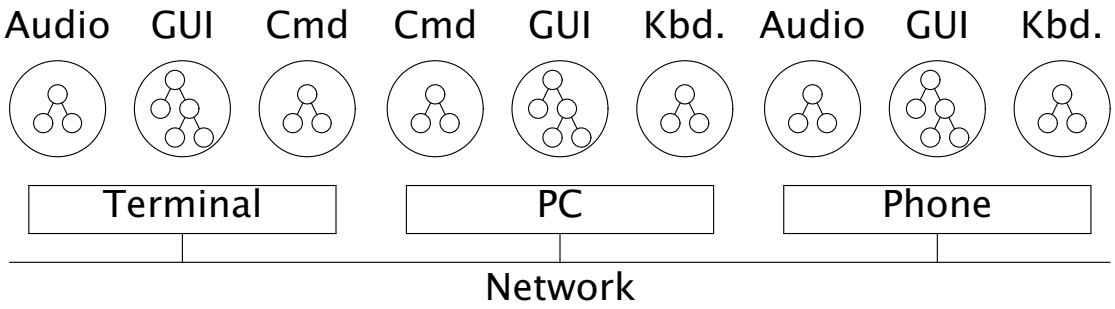
```
void* readf(char*f, void* buf, long* l);
long  writef(char*f, void* buf, long l);
```

New flags for existing calls

New drivers and programs

System architecture

A system built out of dynamic vols.



System architecture

Resource volumes

- Exported file trees
- Packaged with constraints
- Announced to the network
- Expected to come and go.

System architecture

Components:

- Volume device
- Volume daemon
- Dynamic mount table
- get-it-all, put-it-all calls.
- The common idiom is 9P

System architecture

Types of systems

- Plan B nodes export/use volumes
- 9P speakers may use volumes
- 9P speakers may export volumes
- Applications run in Plan B nodes
- Files can be gatewayed

Resource volumes

Volume daemon

- Keeps most of it out of the kernel
- Runs the discovery protocol
- Mounts remote volumes
- Tells the kernel

Resource volumes

Volume device

- Updated by the user (or daemon)
- Keeps the kernel aware of vols
  - Volume table: name, cnstr, dom, mnt.
  - volume kernel interface
- Let's the user know which vols.

Dynamic mounts

Plan 9 is no longer Plan 9

- Plan 9 users mount their resources
- Plan B users ask for them
- Plan 9 name spaces are static
- Plan B name spaces are dynamic

Dynamic mounts

Implementation sketch

- Plan 9 NS:
  - Table of file → file...
- Plan B NS:
  - Old file → file...
  - New file → mvol...

Dynamic mounts

Implementation sketch

- An mvol is a insertion point
- Prior to using an NS
  - The kernel checks for vols
  - and might rewrite the NS
- The FD table is kept untouched

System interface

Kept as it was, plus...

- readf/writef/createf calls
- Important FSs replaced: cmd, ui, ...
- new MVOL/MVOLUN mount flags
- Boot procedure changed

Current status

The system is operational

- All Plan 9 apps run well
- New ones are being added
- The kernel is still experimental
- But we use the services daily

Questions?

More information via our httpd

`http://lsub.org`

or plan B's experimental httpd

`http://b.lsub.org`

...we call our logo "The fish"