

# PhantomOS

## Implementation Summary

*"To Create, Not To Destroy"*

### Core Philosophy

PhantomOS is an operating system where destruction is architecturally impossible. There are no rm, kill, or delete commands. Data is hidden, not deleted. Processes become dormant, not terminated. All history is preserved in the geological record.

## 1. GeoFS (Geology FileSystem)

Files: geofs.c, geofs.h

### Key Features:

- Content-addressed storage: Files stored by SHA-256 hash
- Append-only: Data is never overwritten, only new versions created
- Views: Time-travel through storage history (like geological layers)
- References: Paths map to content hashes
- Hide, not delete: Content remains in geology, just hidden from view

### API:

```
geofs_volume_create/open/close()    // Volume lifecycle
geofs_content_store/read()          // Content operations
geofs_ref_create/resolve/list()     // Path references
geofs_view_create/hide()            // View management
```

## 2. VFS (Virtual File System)

Files: kernel/vfs.c, kernel/vfs.h

POSIX-like VFS layer that routes operations to different filesystem backends. Supports multiple filesystems (procfs, devfs, geofs), file descriptors, directory operations, symlinks, and hide() instead of unlink().

### Mounted Filesystems:

Mount Point	Type	Description
/proc	procfs	Process information
/dev	devfs	Device files
/geo	geofs	Persistent storage

## 3. Procfs (Process Filesystem)

File: kernel/procfs.c

Virtual filesystem exposing process and kernel information:

/proc/version, /proc/uptime, /proc/stat, /proc/constitution, /proc/mounts, /proc/{pid}/status

## 4. Devfs (Device Filesystem)

File: kernel/devfs.c

### Devices:

Device	Description
/dev/null	Discards writes, returns EOF on read
/dev/zero	Returns infinite zeros
/dev/random	Returns random bytes
/dev/console	Terminal output

## 5. Phantom Kernel

Files: kernel/phantom.c, kernel/phantom.h

### Process Management:

- States: embryo -> ready -> running -> blocked -> dormant (never terminated)
- No kill(): Only phantom\_process\_suspend() and phantom\_process\_resume()
- Persistence: Processes saved to GeoFS, survive kernel restarts

### IPC:

Message passing with mailboxes, stored in GeoFS for persistence. Types: DATA, SIGNAL, REQUEST, RESPONSE

### Scheduler:

Round-robin, priority, or fair-share modes. Priority levels 0-31. 10ms default time slice.

## 6. Init System

Files: kernel/init.c, kernel/init.h

### Service States:

State	Description
SERVICE_EMBRYO	Being created
SERVICE_STARTING	Dependencies resolving
SERVICE_RUNNING	Active and healthy
SERVICE_DORMANT	Inactive but preserved
SERVICE_AWAKENING	Transitioning to running

### Built-in Services:

geofs, vfs, procfs, devfs, governor, shell

## 7. Governor (AI Code Evaluator)

Files: kernel/governor.c, kernel/governor.h

Capability-based + interactive code evaluation system.

### Capabilities (16 types):

Files: read\_files, write\_files, create\_files, hide\_files

Process: create\_process, ipc\_send, ipc\_receive

Memory: alloc\_memory, high\_memory

Elevated: network, system\_config, raw\_device, high\_priority

Info: read\_procfs, read\_devfs

Special: governor\_bypass

### Threat Levels:

Level	Action
NONE	Auto-approve
LOW	Auto-approve with logging
MEDIUM	Prompt user (interactive mode)
HIGH	Prompt user or decline (strict mode)
CRITICAL	Always decline (destructive code)

### Destructive Patterns (~50):

File: unlink, remove, rmdir, truncate

Process: kill(), abort, exit()

System: reboot, shutdown, format

Database: DROP TABLE, DELETE FROM

Shell: rm -, >/dev/

## 8. Shell

Files: kernel/shell.c, kernel/shell.h

### Commands:

Navigation: pwd, cd, ls, stat

File Operations: cat, echo, mkdir, touch, write, ln -s

Phantom-Specific: hide (preserved in geology)

Process: ps, suspend, resume

System: constitution, geology, mount

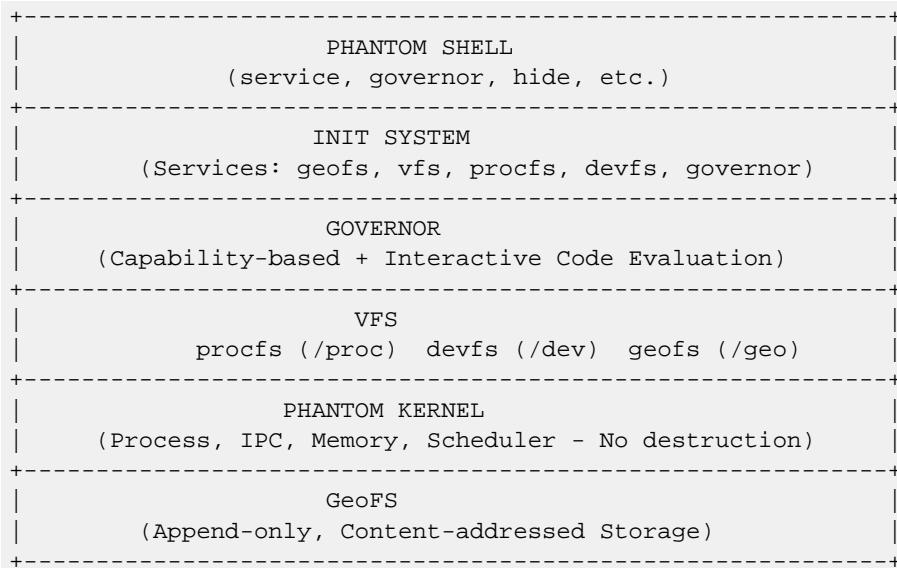
Services: service list/status/awaken/rest

Governor: governor status/stats/mode/test

### Notable Absences:

No rm, rmdir, unlink, kill, killall, delete, destroy

## Architecture Diagram



## File Summary

File	Lines	Description
geofs.c	~1100	Append-only storage
kernel/phantom.c	~1600	Kernel core
kernel/vfs.c	~900	Virtual filesystem
kernel/shell.c	~1500	Interactive shell
kernel/init.c	~800	Init/service manager
kernel/governor.c	~750	AI code evaluator
kernel/geofs_vfs.c	~600	GeoFS-VFS adapter
kernel/procfs.c	~400	Process filesystem
kernel/devfs.c	~350	Device filesystem

Total: ~9,500 lines of C code

*Generated: January 2026*

*PhantomOS - "To Create, Not To Destroy"*