C>ONSTRUCTOR

01.02.2026

Practical Persistence

Prof. A. Tormasov (*CKL, C>T*)

R. Mitov (*CUB, C>T*)

constructor.tech

persistence: the period of time during which an object is usable

M. P. Atkinson, P. J. Bailey, K. J. Chisholm, W. P. Cockshott, and R. Morrison, “Ps-algol: A language for persistent programming,” (1983)

Your programs should not fear death!

The Traditional Way...

Persistence managed
by the application!

PhantomOS

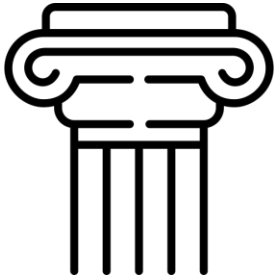
A non-linux OS, offering orthogonal persistence!

The Principles of Orthogonality

A Userspace Programmer's Perspective

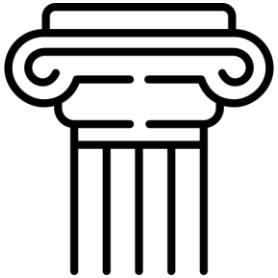
Independence

All data treated the same, despite longevity.



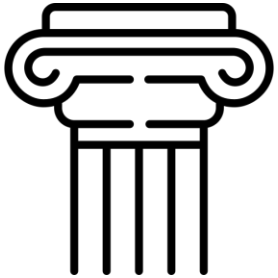
Data Type

All data should be persistent.



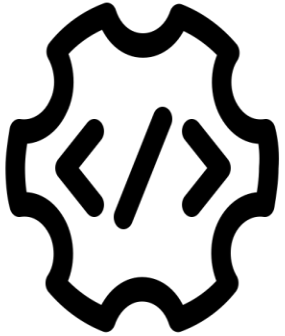
Identification

Type system should not influence data persistence.

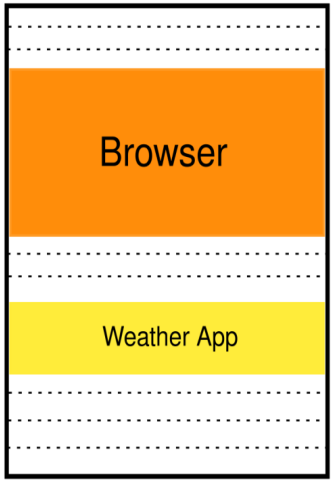


The userspace programs are not responsible for their persistence… the kernel is!

PhantomOS: Under the Hood

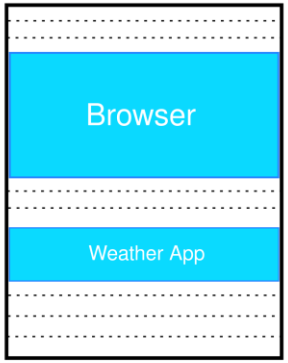


Phantom Virtual Machine



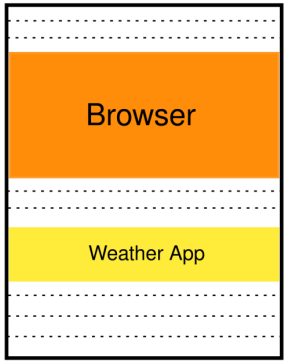
Persistent Address Space

PhantomOS: Under the Hood



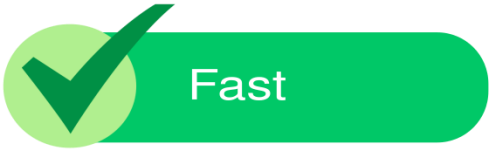
Frozen Persistent Address Space

"Superblock"



Persistent Address Space

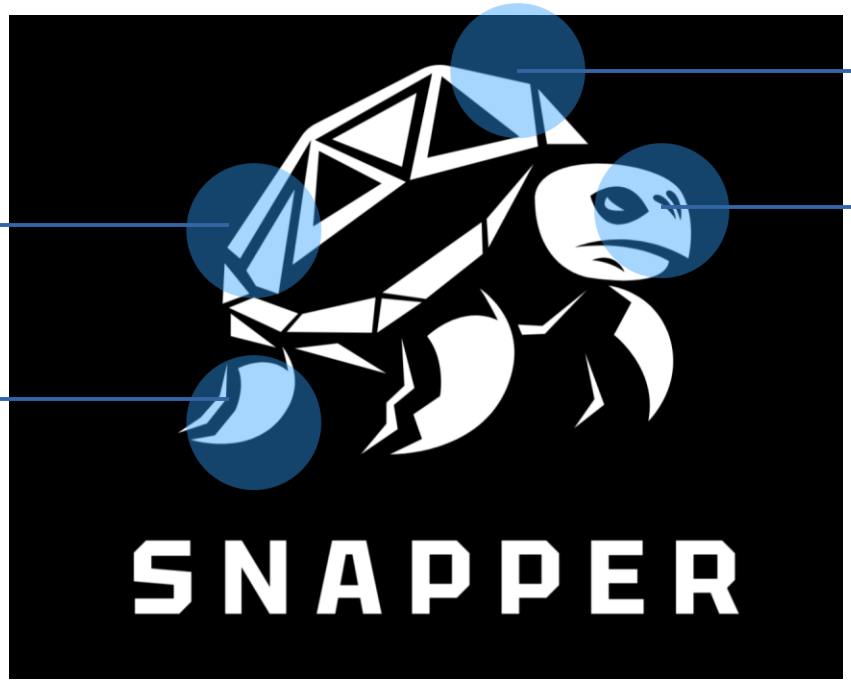
The “Superblock” Approach



Enter the Snapper!

File-System Support

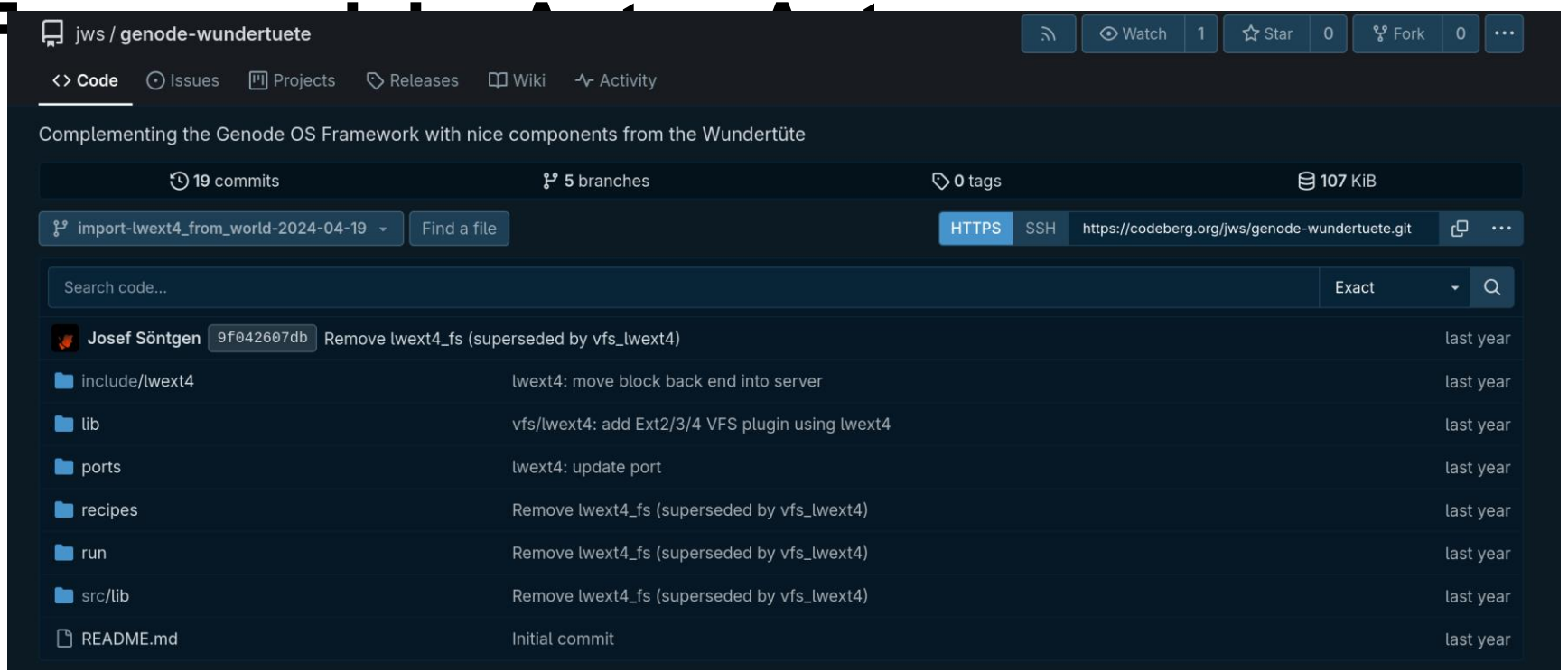
Retention & Redundancy Policies

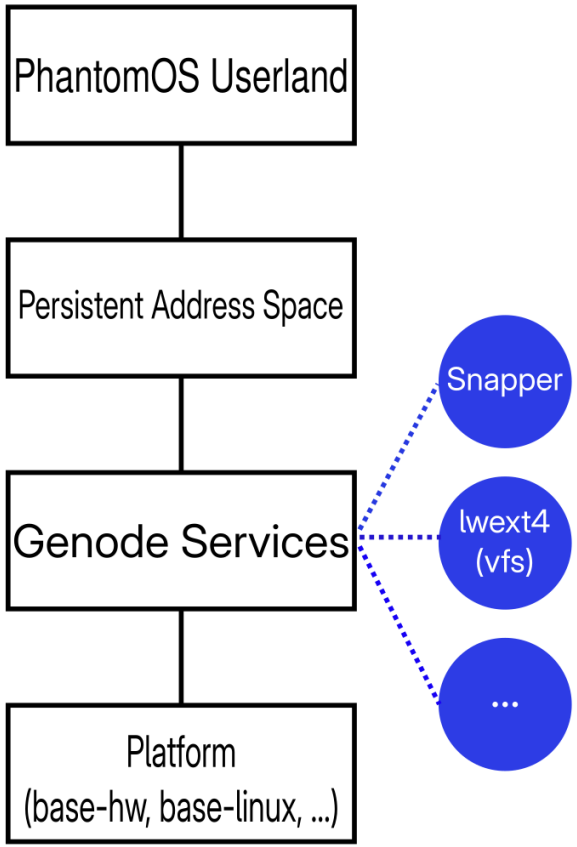


Pause-able Snapshots

Multiple Generations of Snapshots

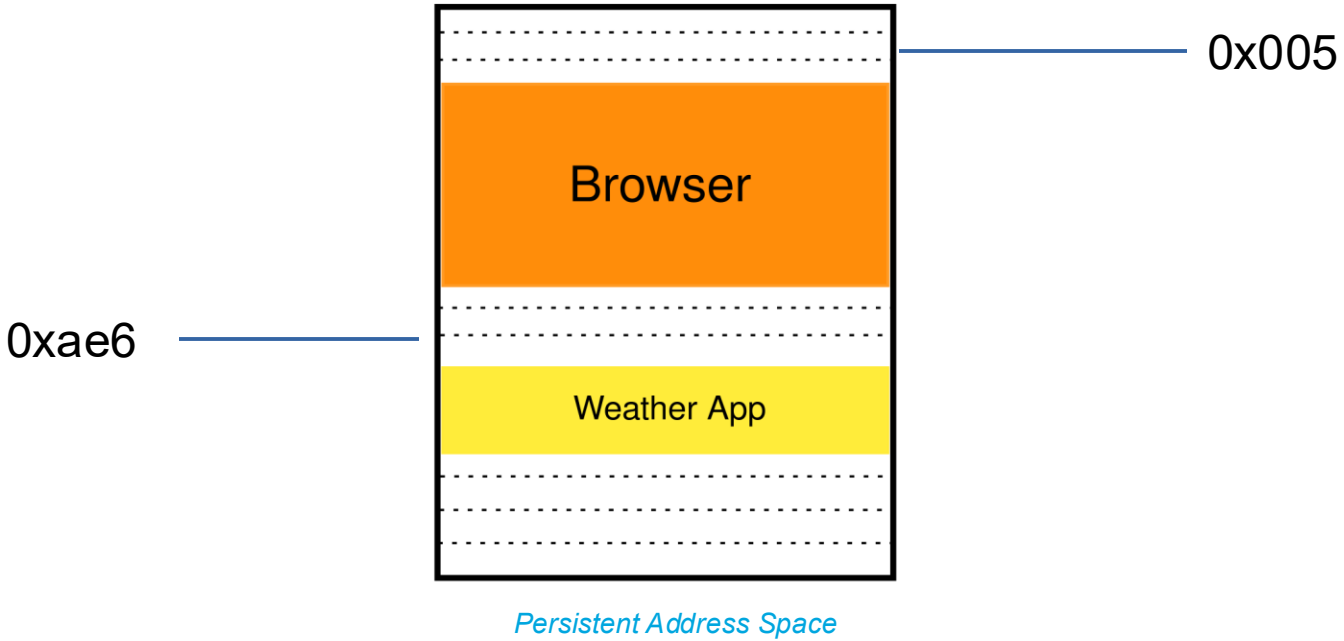
PhantomOS ported to Genode



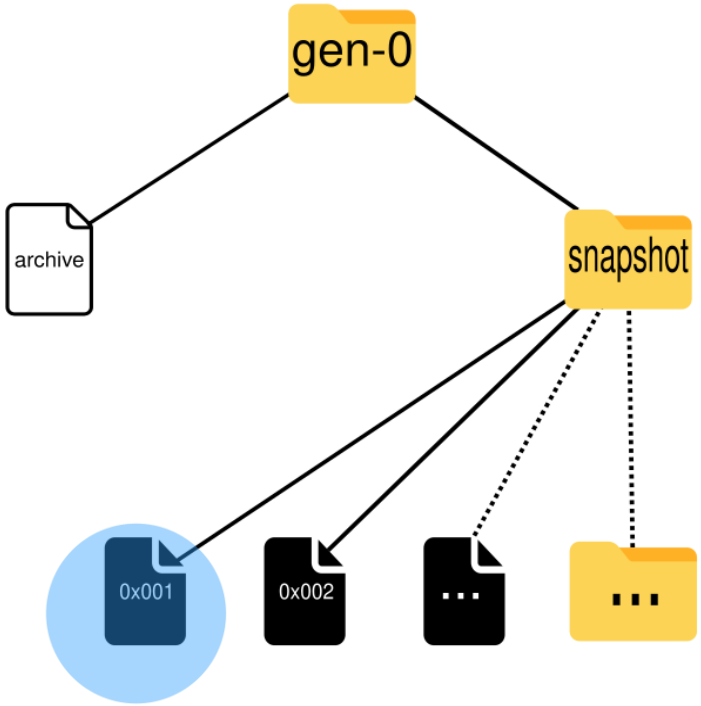


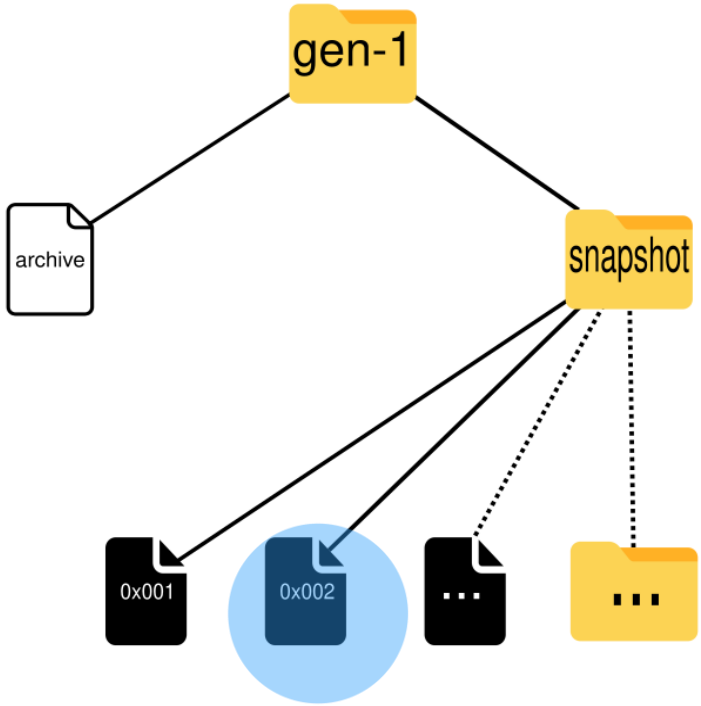
System Architecture

Snapper: Inner Workings



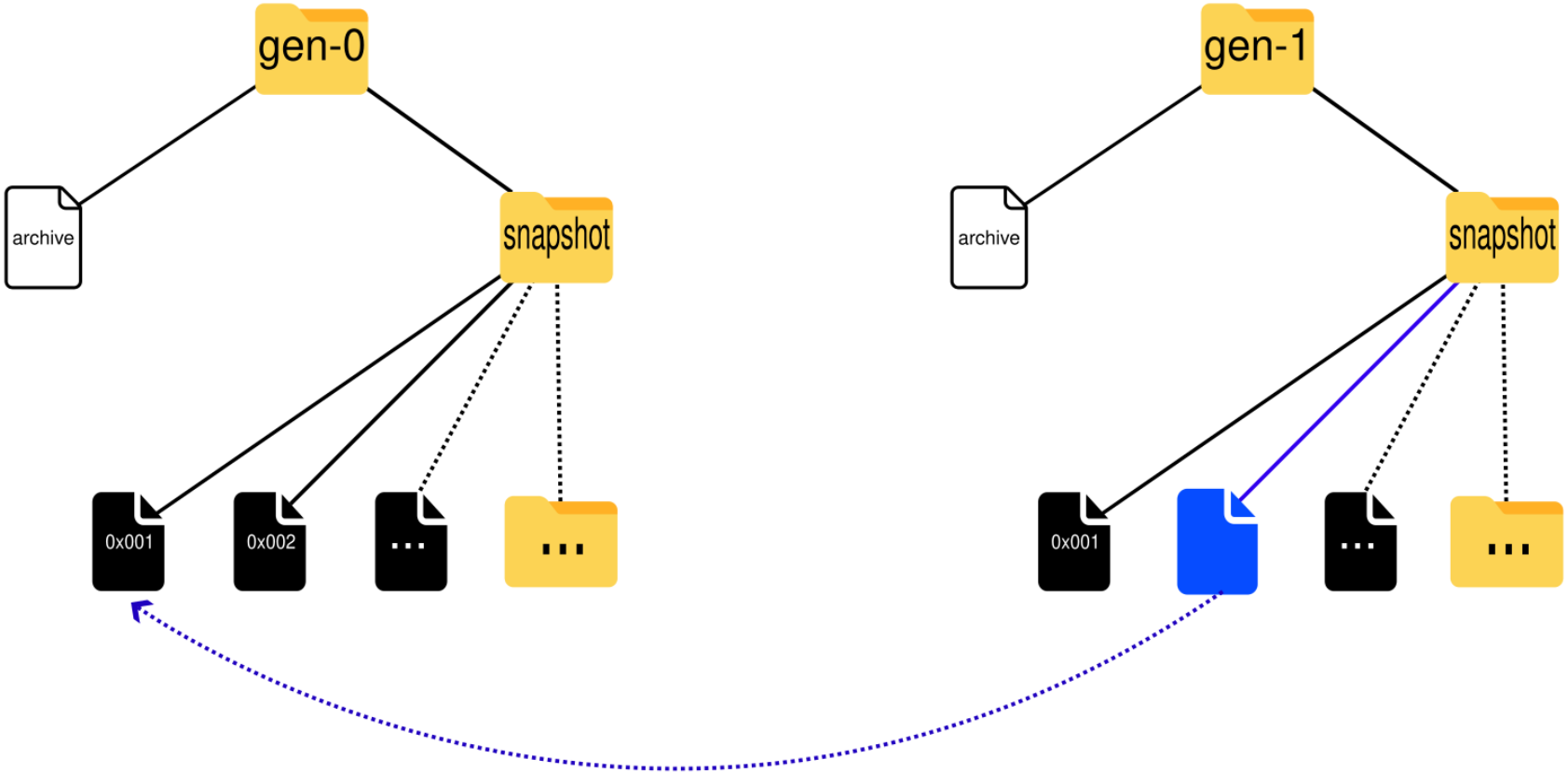
Snapper: Inner Workings





Same file?

Snapper: Inner Workings



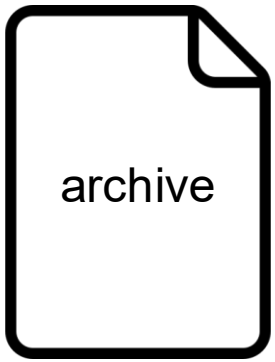
Snapper: Inner Workings

Snapshot File

```
+----+---------------+----+---------...
| v  | hash           | rc | data
+----+---------------+----+---------...
```

Symbol	Size	Description
v	1 byte	Snapper version
hash	4 bytes	integrity for the data
rc	1 byte	reference count (unsigned)
data	as required	the snapshot content

Snapper: Inner Workings



Archive File

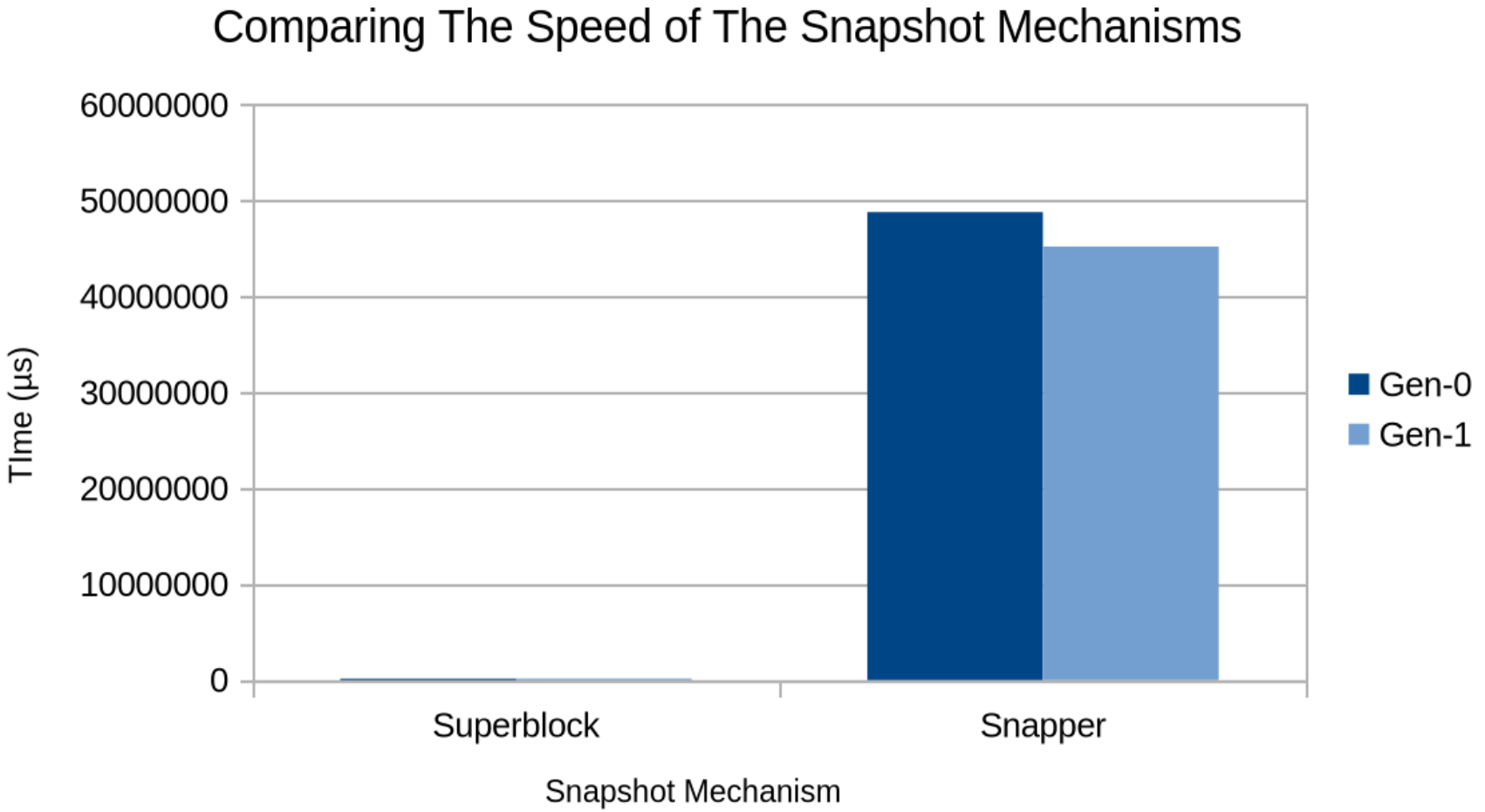
```
+----+---------------+--------------------------------+---------...
| v  | hash           | n                              | data
+----+---------------+--------------------------------+---------...
```

Symbol	Size	Description
v	1 byte	Snapper version
hash	4 bytes	integrity check for the data
n	8 bytes	number of entries in the data
data	as required	key-value map that contains snapshot files' paths

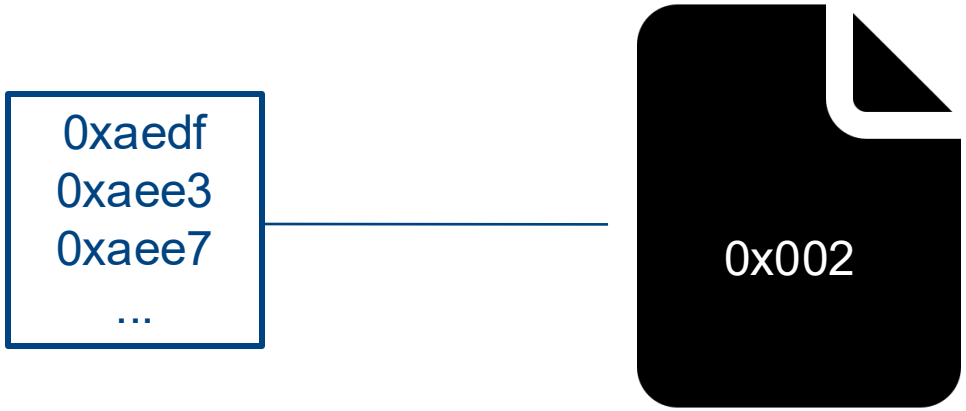
Benchmarks!

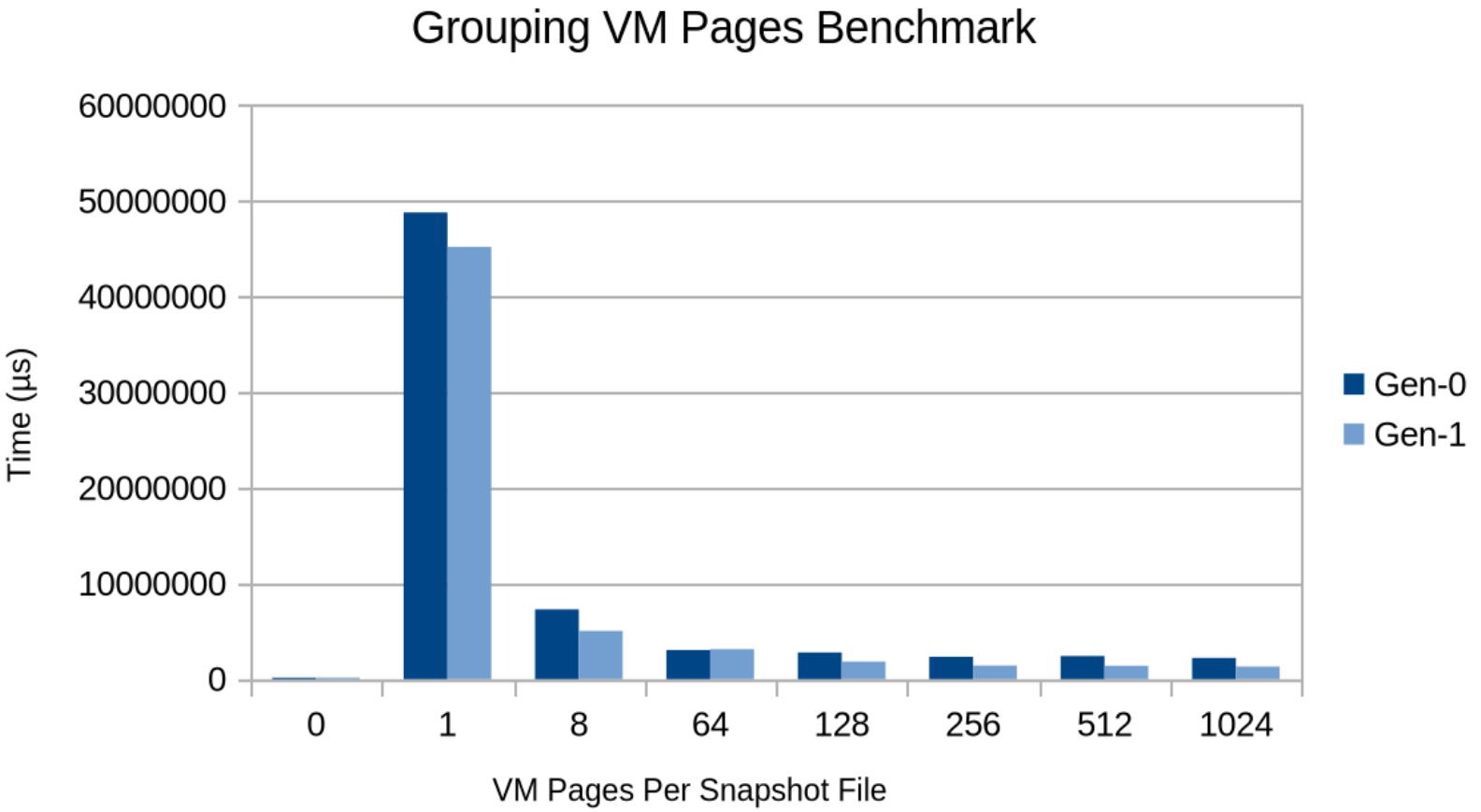
CPU: AMD Ryzen 7 6800H with Radeon Graphics
Platform: x86, base-hw (QEMU)
Resources: 12 GB RAM, 8 cores
Build: Graphics enabled
Commit*: https://github.com/rumenmitov/phantomuserland-snapper/tree/b44116b3b32a75eee901205131a69874d0f633be

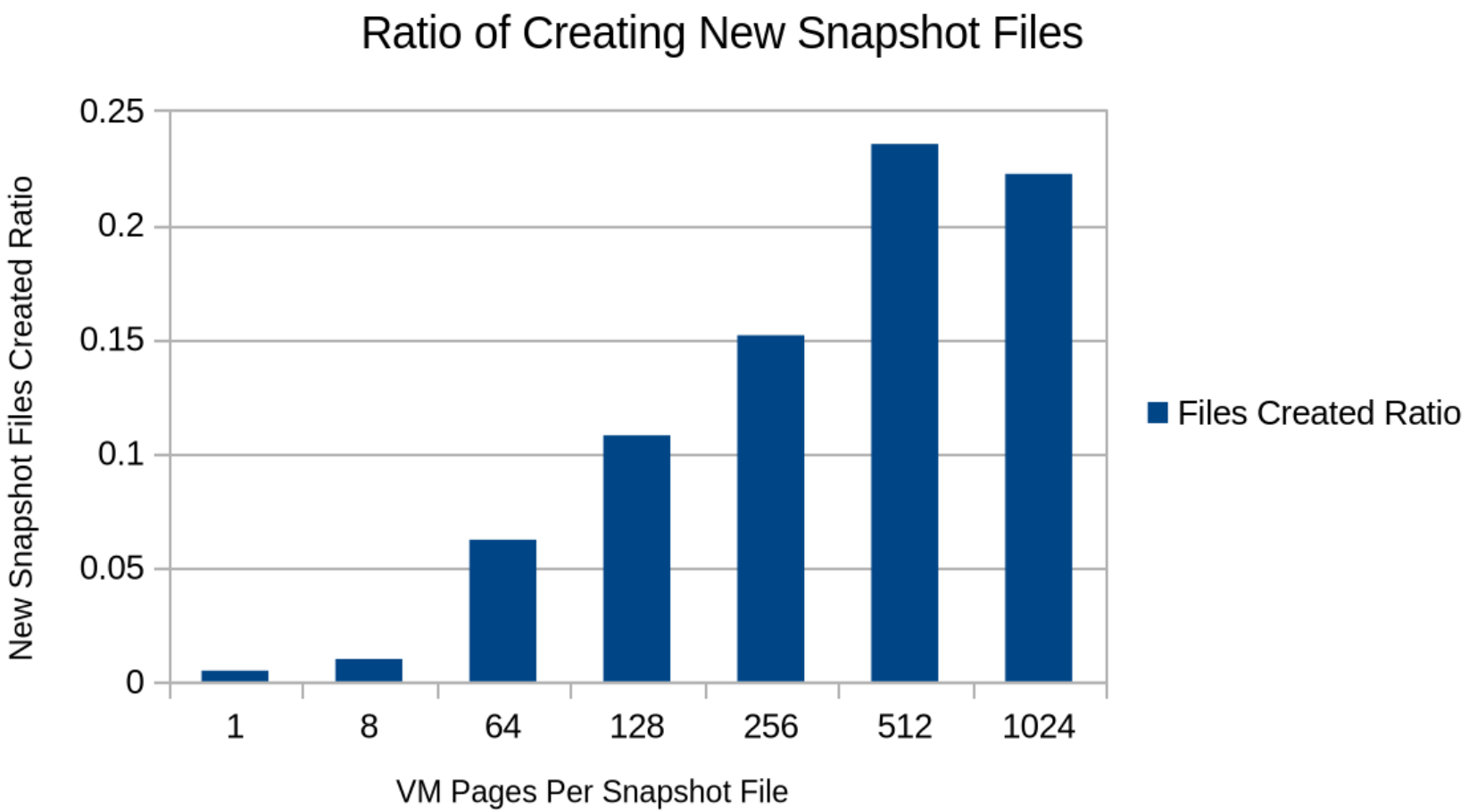
*commit does not contain the benchmarking code


Comparing The Speed of The Snapshot Mechanisms
60000000
50000000
40000000
30000000
20000000
10000000
0
TIme (µs)
Superblock
Snapper
Snapshot Mechanism
Gen-0
Gen-1


0xaedf
0x002


0xaedf
0xaee3
0xaee7
...
0x002


Grouping VM Pages Benchmark
60000000
50000000
40000000
30000000
20000000
10000000
0
Time (µs)
0
1
8
64
128
256
512
1024
VM Pages Per Snapshot File
Gen-0
Gen-1


Ratio of Creating New Snapshot Files
New Snapshot Files Created Ratio
0.25
0.2
0.15
0.1
0.05
0
1
8
64
128
256
512
1024
VM Pages Per Snapshot File
Files Created Ratio

Snapper: Configuration

```
<start name="snapper" ram="500M" caps="300">
  <provides>
    <service name="Snapper"/>
  </provides>

  <config
      verbose="true"
      redundancy="2"
      max_snapshots="5"
      min_snapshots="2"
      threshold="1000"
      expiration="3600"
      bufsize="10M">

    <rtc/>

    <vfs>
      <fs/>
    </vfs>
  </config>
</start>
```

Fault Tolerance FAQ

Q: System shutdown during snapshot process leaves file-system in a broken state!

A: Use the **fsck** utility to fix the file-system.

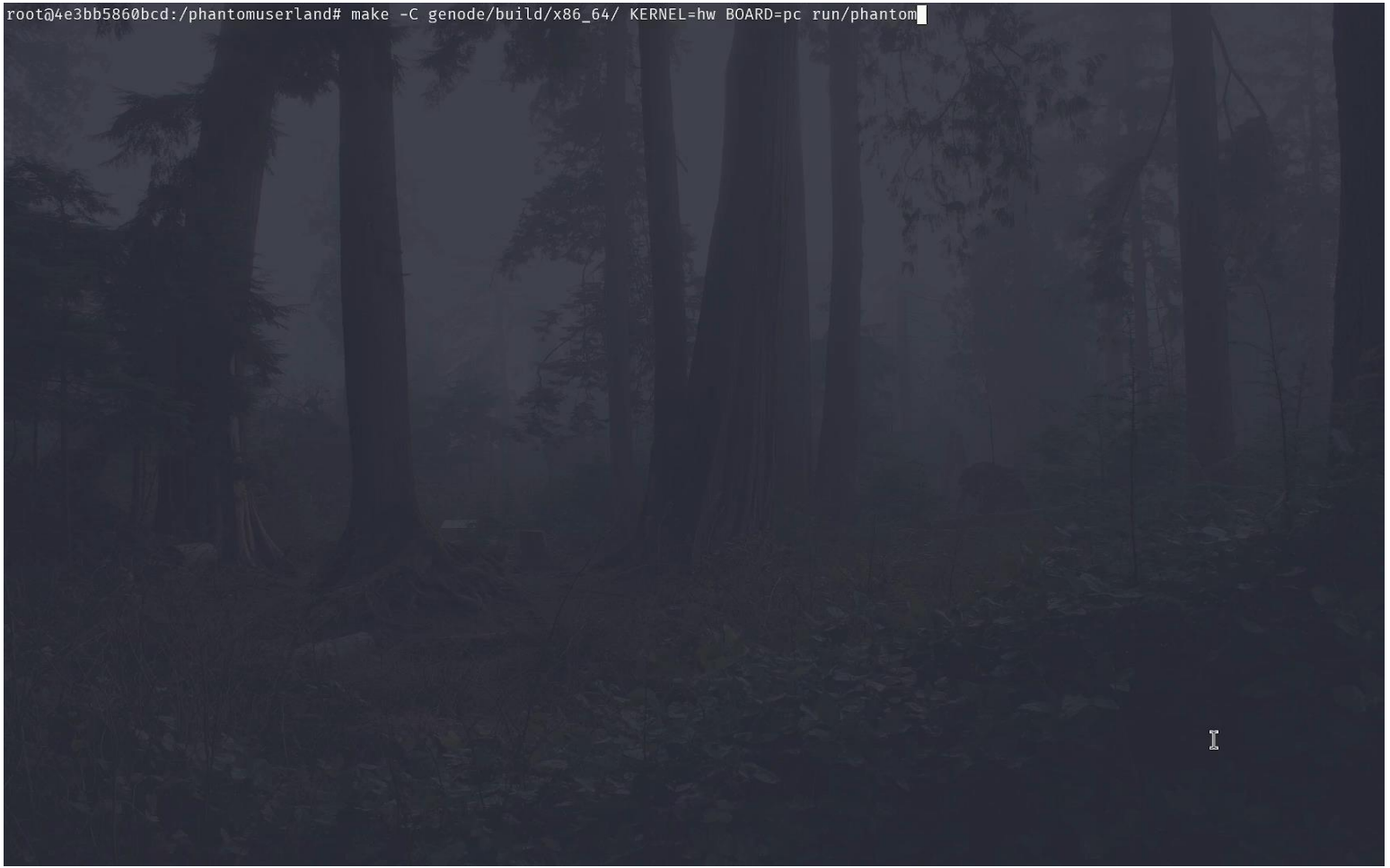
Q: After using the fsck utility, Snapper complains that some of the snapshot files have invalid hashes!

A: The files were unable to be properly saved at the time of the shutdown. They cannot be recovered and must be manually **removed** to stop the warnings.

Other PhantomOS Developments

WASM runtime port by Kirill Samburskiy

Persistent networking by Anton Brisilin

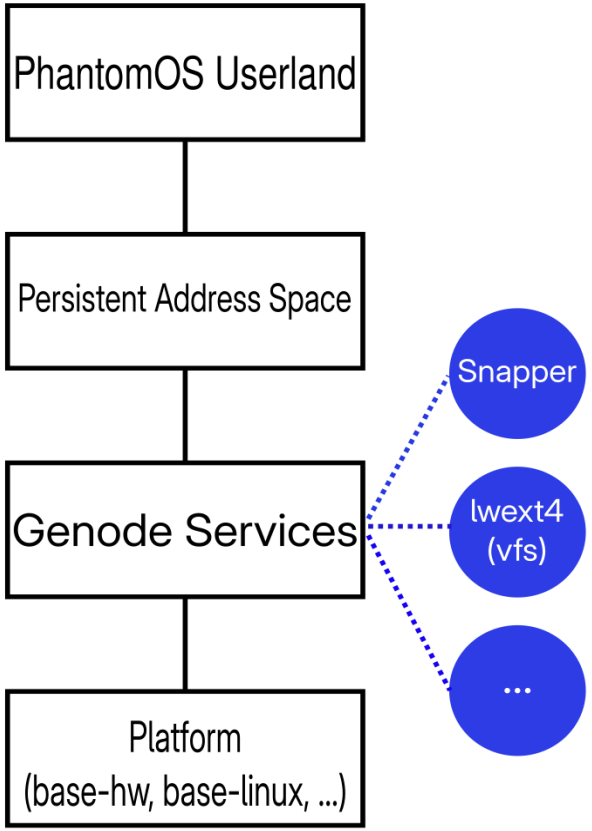

root@4e3bb5860bcd:/phantomuserland# make -C genode/build/x86_64/ KERNEL=hw BOARD=pc run/phantom

Practical Persistence

Are traditional persistence mechanisms outdated?

PhantomOS's "superblock" approach

Disk storage fault tolerance with Snapper



PhantomOS 2020 FOSDEM Talk: https://archive.fosdem.org/2020/schedule/event/uk_phantom/

Snapper Repo: https://github.com/rumenmitov/snapper

PhantomOS Repo (with Genode & Snapper): https://github.com/rumenmitov/phantomuserland-snapper