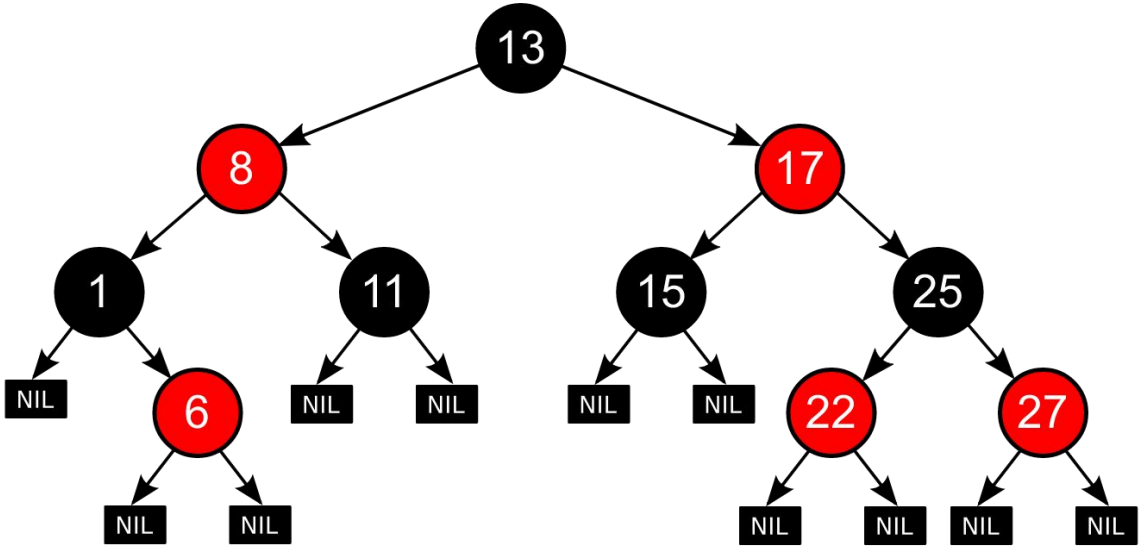
Team 9 - Fin-Dows

François, Marc, Luca, Georgijs

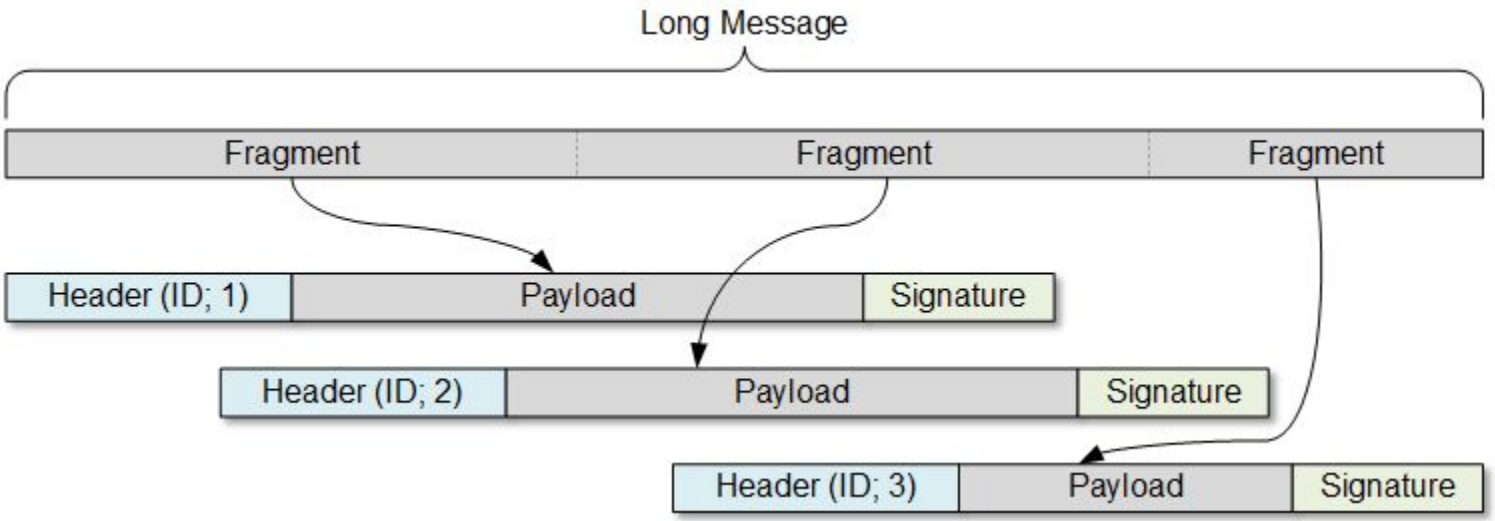
Virtual Memory: Red-Black Tree

- We implement a **Red-Black Tree** to efficiently handle (de)allocation of virtual address space



Message Passing: Fragmentation

- LMP and UMP have fixed **maximum message size** (around 64 bytes)
- We support **arbitrarily large messages** through **fragmentation**



Message Passing: Single-Threaded Init Process

- Init main thread is running **event loop** all the time
- **No other threads** are ever spawned
  - All operations must be **non-blocking**
- `async_channel` is our abstraction for communicating between cores
  - Allows **multiple concurrent requests**

Filesystem

- Read and write from a fat32 filesystem
  - Implemented in **fat32.c**
  - Tested with **unit tests** & **rpc tests**
- We can also execute an ELF image in the filesystem
  - We can load from the **sd card**
- By design, the filesystem is ***thread safe***

Shell

- Custom Readline Library
  - Implemented using **gap_buffer** (**dynamic_array**)
- Session Variables, Syntax Highlighting, Tab Complete, …
  - Implemented using **trie**
- Support Piping I/O between processes
  - Shared UMP between processes (Performance!)
  - Pipe into a files (via tee)
- List of Commands (+ **short-circuiting**):

```
1 ~$ true && echo "short-circuiting example"
2 short-circuiting example
3 ~$ false && echo "short-circuiting example"
```

Networking

- Fully asynchronous implementation
  - **Asynchronous** communication between the network driver, network handler and processes
- IP resolving using ARP request
  - Cache IP-MAC using a **hashtable**
- ICMP support for ping requests
  - **Timeout** support
- UDP support
- Remote shell support using UDP

```
Network Commands
  ping            ping IP address                                 [alias]
  send            send UDP packet                                 [builtin]
  listen          listen on some port                             [alias]
  setio           set io method                                   [builtin]
```

Distributed Capabilities

- Support the **full set of capability operations** (found several bugs in the kernel and distops library)
- **Capability transfer system** is integrated into communication channel between cores (async_channel)
  - Rest of system can **transparently** send capabilities anywhere

One Command to Rule Them All

The following command showcases all system functionality at once:

```
run /sdcard/ping -c 5 10.0.3.0 | oncore 1 wc > /sdcard/out.txt ; cat /sdcard/out.txt
```

- **Load** program from the file system
- **Read** packets from network
- **Pipe** output to program on core 1 using shared UMP frames
- **Pipe** output back to core 0 and write to file system
- **Read** from file system

Thank you!

Any questions?