Creation

* Google created the open-source OS Fuchsia in 2016
* Inspired by web browsers general function
* 2 projects called “Pink + Purple = Fuchsia”
  + Pink: was a code name given by the early Apple and IBM engineers to a new operating system. They chose this name because they used to jot down ideas on blue and pink index cards. The **pink cards** represented technologies for a future dream system.
  + Purple: was a code name given to the prototype devices that would later be released to the public as the first iPhones.

General about

* Zircon kernel
* Parallel to linux’s way of operations, without building off the linux kernel
  + Similar hierarchy for thread handling (More later)
  + Operation of handles, memory virtualization tactics, and use of threads are all very similar which will also be discussed later
* Unlike many languages, based off a separate micro kernel
  + Zircon Kernel
  + Typical kernels IPC, memory management, scheduler, process management, user permissions, networking, file systems, and device drivers all come stock
  + This kernel only has very limited code, increasing security, efficiency, scalability
    - Only has IPC, memory management, and scheduler, process management
* Built for ARM64 (recommended), Intel x86-64 (using Rosetta 2), Windows not supported

Processes/threads

* Created using zx\_process\_create command
  + Handles are in place of a function pointer involving void pointers
* Jobs are capable of running multiple processes at the same time
* Each process then has one/many threads associated with it capable of running
  + Jobs > Processes > Threads
* Process are containers for handles, virtualization addresses, and threads
* These handles then serve as a connector between processes and kernel objects
* “To create a process, a handle is made that moves into the kernel space and creates a kernel object which is then linked back to the process in the user space with another handle.”

Memory/Virtualization use

* Very similar to many common day languages
* Utilizes Virtual Memory Address Regions (VMRA)
* Both kernel space and user space are capable of utilizing memory in this manner

Concurrency

* Zircon uses Futex
  + Fast User space Mutex
  + No kernel rights involved
* Wait and Wake system for concurrency
* Differences from Linux Mutex
  + Linux uses a very physical memory address space Mutex
  + Zircon uses a virtualized version of mutex
    - Zircon Futex are separated from the kernel
    - These Futex have no access to the kernel space

Notes

* No root permissions
  + Utilizes namespaces instead
  + If you have correct namespace, you are “root” for that system
  + Have to stay within your own namespace
    - Don’t have root access to other namespaces

Advantages

* Can run flutter applications
* Has lots of potential for growth
* Lightweight OS

Disadvantages

* Still in development
* Applications are not widely supported
* Needs an emulator