0 - std. input
1 - std. output
2 - std. errors
files

./ - curr. dir
../ - parent dir
/ - the root

system calls - low level
system libs - wrap

PIPE - only between child proc.
- stored in a buffer in mem.

FIFO - can com. between programs
- files on disk

PROCESSES vs THREADS → only creates a stack in memory

pure area - does not get copied.
impure - var + heap + stack
↳ copies the main proc. after fork()

⊕ save memory.
⊕ share data easily
⊖ has no security.

Threads ⇒ mutex
↳ lock
⇒ barrier ⇒ waits for n threads
⇒ semaphore ⇒
how many th. can execute a inst simult.

Zombie proc. - proc. that has compl. exec. but is still in the proc. table.
Orphan: child alive, parent dead.

read: w8 for data or no writer connected
write: w8 for space or no reader connected
*open: w8 for open in both ways. (complem. OP)

Symbolinks - „short-cuts"; reference to another file
- if deleted: points to nothing.

Hard links - created only by the root
- exact replica of the file
- if deleted: links still working having acces to data

Deadlocks - Get out: choose a victim and stop the proc victim.
• go back to a check point
- detect one: look for cycles.
- prevent: keep the same order for opening / locking resources.
- Causes: mutual exclusion.
↳ resources that wait for each other.

Semaphore — (V(s), C(s)
value of the sem ↓ ↳ waiting list
↳ pointers

Mounting = associate a storage device to a particular loc. in the dir tree.

Cadus — Direct cache
+ fast and simple
- may lead to ~~thr~~ cache ~~thr~~ trashing (collision)

— Associative cache
- place on each page on a free location. found by iter. through the cache
- rather slow

- Set associative cache:
- hash table
- the cache is org. in sets of pages.
- the cat of a page is comp. on pg. address % cache set.
- flexible; avoid collisions.

Proc states: Hold: - becomes a proc. but not executed
- held back by the job. planning. and resource allocation.

Ready: loaded in memory but doesn't have a processor to execute it

Run: oscillates between run, ready

Wait: when doing I/O

Swap: when memory full

Finished: resources are dead process done.

Process Scheduling

- First come, first served
- small proc. waits after the big ones.
- shortest job first
- shortest job smash in
- Priorities
- by lenght, deadlin

Round Robin.

Allocation methods
- single
- Multiuser — Fixed partitions
  - Absolute
  - Relocatable
  - Variable partitions.
- Virtual
  - paged - mem. calculation (add.
    - slow;
  - Segmented - reusable segment
    - protected mem. access
  - paged-segmented.

Fragmentation.

Choose a victim

FIFO; Not recently used (2 bits); Least recently used matrix