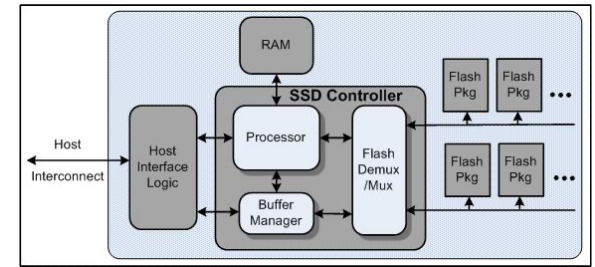
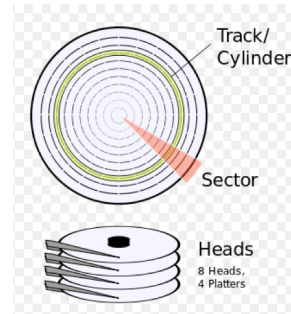


> Solid State Drives/Solid State Disks (SSD)s

NAND Flash. Use quantum tunneling through an insulator to write/erase! Limited number of erasures. No moving parts.

Requires separate controller:

- Error-correcting code (ECC), Bad block mapping
- Block erasing, Wear leveling
- Read and write caching, Garbage collection, Encryption



Very fast random access & throughput. Can be limited by bus speeds.
e.g. SATAIII speeds 6Gbit/s (600MB/s bus)

Benchmarks measure 'IOPs'

SLC (Single bit per cell); 100K writes per cell
MLC Multilevel cell (2bits); 10K writes per cell
TLC Triple level (3bits per cell); 3-5K writes per cell, now QLC...

> Spinning disks

Cylinders. Platters. Heads.

Two common rotational speeds
5400 & 7200rpm.
7200rpm = _____ revolutions per second

How many milliseconds for one revolution ? _____

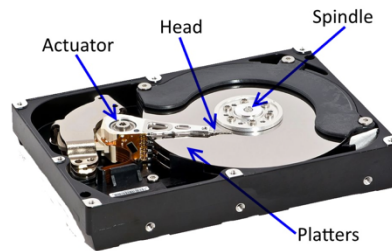
Access time?

Average seek time ~ 10ms (but seeking to next track ~ 1ms).
Average rotational latency.

Tiny contributions:

Command processing time (0.003ms)
Head settling time (0.1ms)

IOPs?



> Signals

For more information man -s7 signal

Can signals be queued?

Signal terminology.

Generated

Pending

Blocked

Delivered

Caught

Disposition

Signal disposition per thread or per process?

Signal disposition after fork?

... after exec?

What is signal masking ?

When would I use sigprocmask ?

When would I use pthread_sigmask ?

So which thread will get the signal?

What are Pending signals ?

From man -s7 signal

"A child created via fork(2) initially has an empty pending signal set; the pending signal set is preserved across an execve(2)."

Sending signals?

pthread_kill

raise

kill

Catching signals ?

sigwait

signalfd

signal

sigaction

Demo: Write a program that demonstrates sigprocmask to block and then unblock a signal.

What's wrong with signal ?

How do I use sigaction ?

```
int sigaction(int sig, struct sigaction *act, struct sigaction *oldact);
struct sigaction {
    void (*sa_handler)(int);
    void (*sa_sigaction)(int, siginfo_t *, void *);
    sigset_t sa_mask;
    int sa_flags;
};
```

```
struct sigaction sa;
sa.sa_handler = handler;
sigemptyset(&sa.sa_mask); //Also sigfillset
sa.sa_flags = SA_RESTART;
/* ^^^ Restart functions if interrupted by handler */
sigaction(SIGINT, &sa, NULL)
```

How do we complete and fix this code to catch SIGCHLD ?
(hint: WNOHANG and a while loop will be useful here)

```
int dezombify(int signal) {
    int status;
    pid_t child;
    child = waitpid(-1, &status, _____);

}
```

```
struct sigaction sa;
sa.sa_handler = _____;
sig_____ (_____);
sa.sa_flags =
sigaction(_____, &sa, NULL)
```

How do I set a threads mask? Why would I want to?

```
int pthread_sigmask(int how, sigset_t *set,
                    sigset_t *oldset);
```

What happens to the new thread during *pthread_create* to pending signals and the thread's signal mask?