CS241 #34 – Disks & Signals

> **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

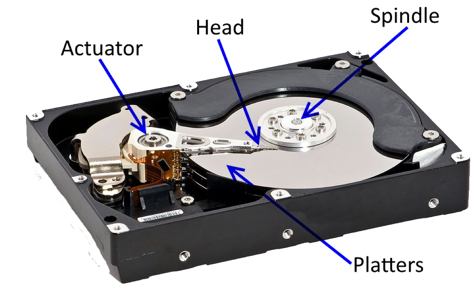
Read and write caching, Garbage collection , Encryption

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); consumer grade; 3-5K writes per cell



**> 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.

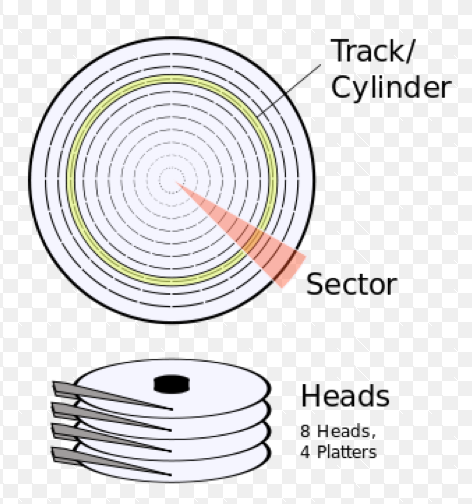
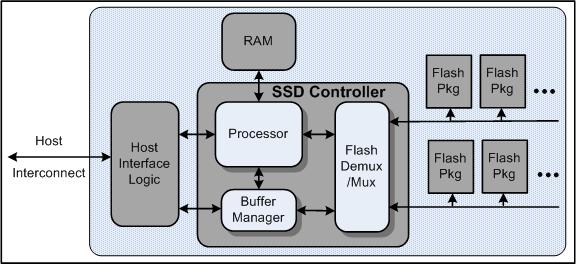
Tiny contributions:

Command processing time (0.003ms)

Head settling time (0.1ms)

IOPs?

Image attrib:/www.usenix.org/legacy/event/usenix08/tech/full\_papers/agrawal/agrawal\_html/index.html



**> 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 SIGCHILD ?  
 (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?