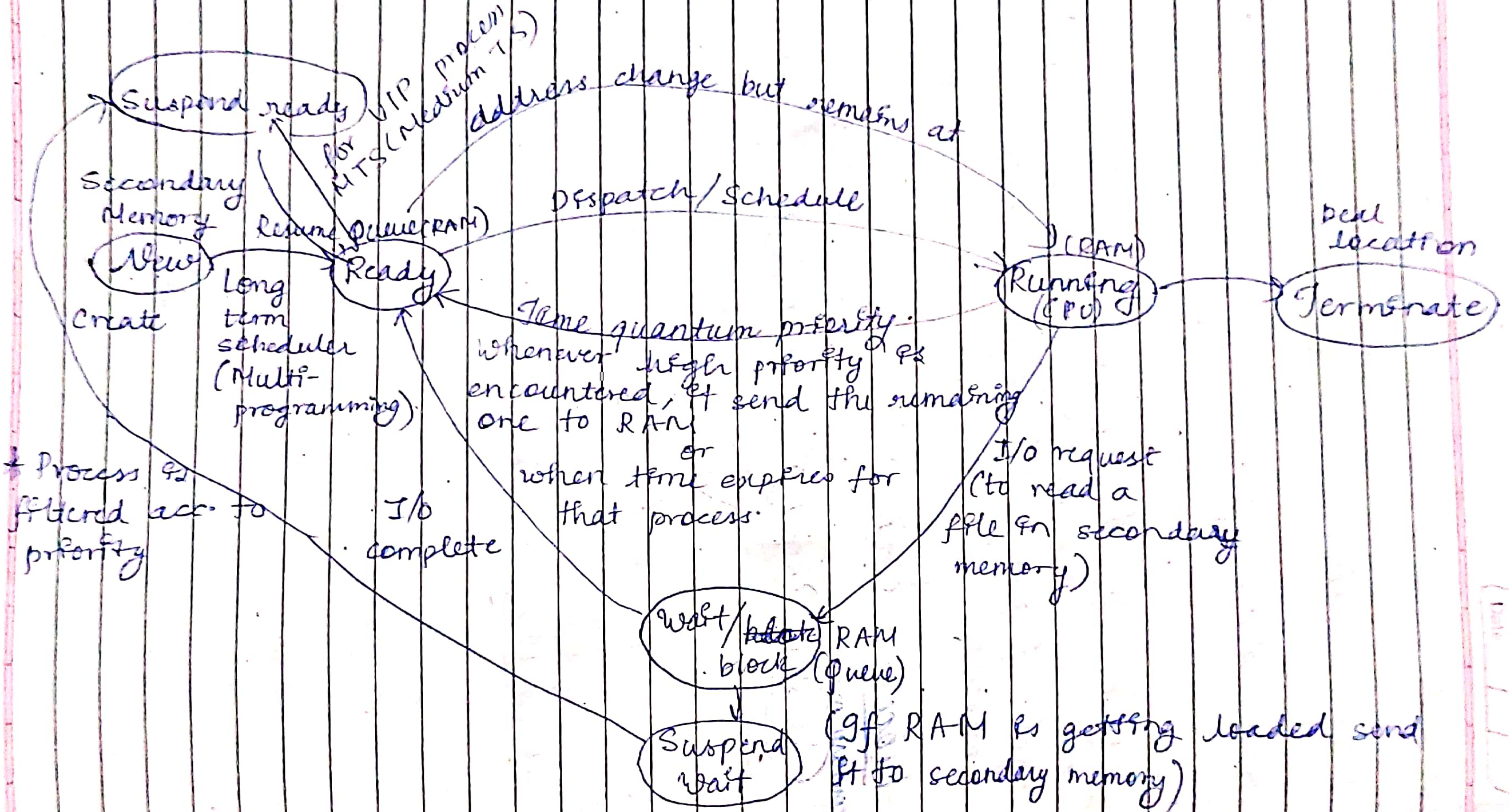
(STS)
Short term scheduler



whenever high priority is encountered, it send the remaining one to RAM or when time expires for that process.

I/O request (to read a file in secondary memory)

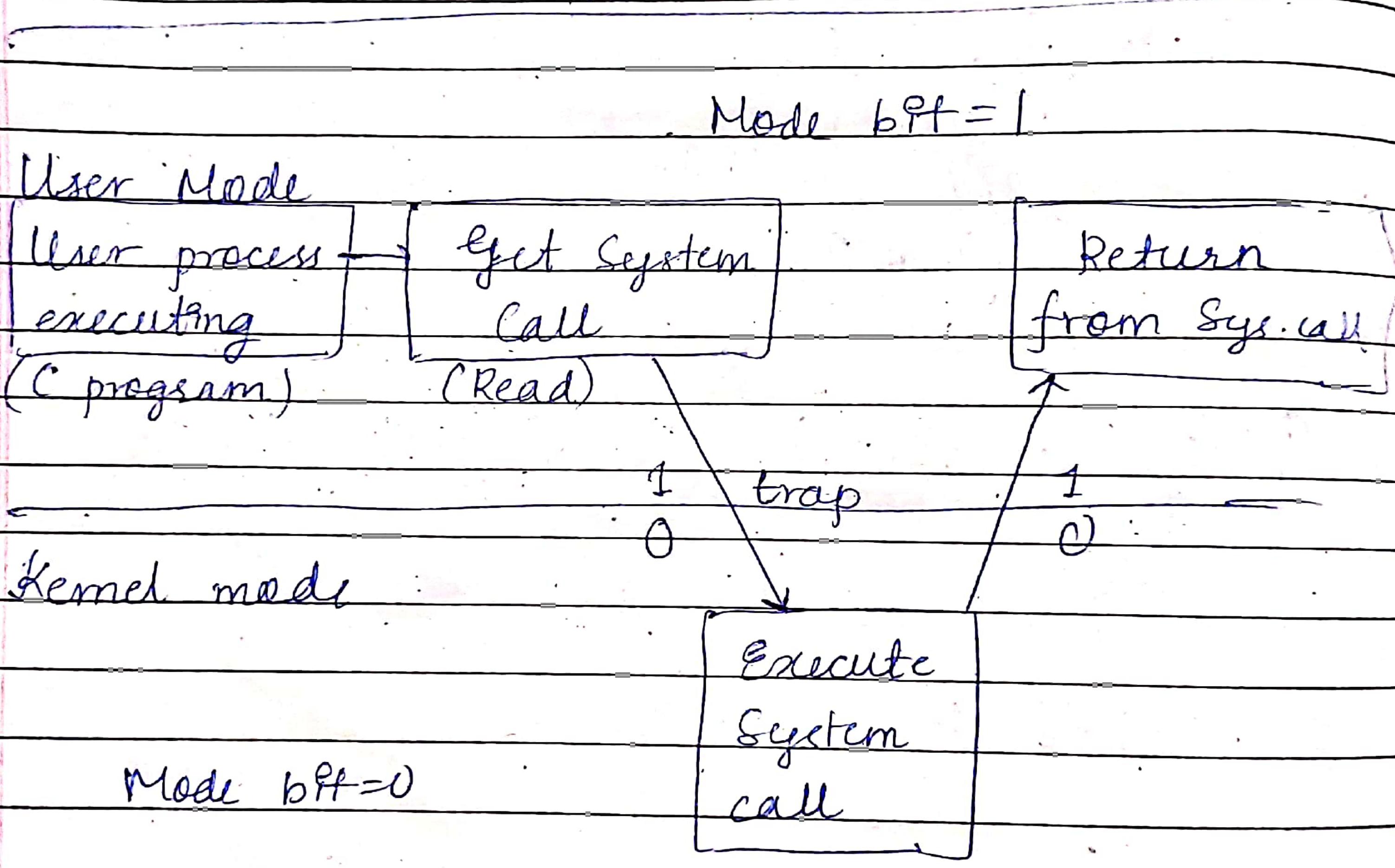
* Process is filtered acc. to priority

(If RAM is getting loaded send it to secondary memory)

User Mode & Kernel Mode

Kernel is the central component of OS that manages operations of computer & hardware- It basically manages operations of memory & CPU. time

HD



1 ⟶ 0

bank ⇌ window ⇌ memory

user mode   Kernel mode

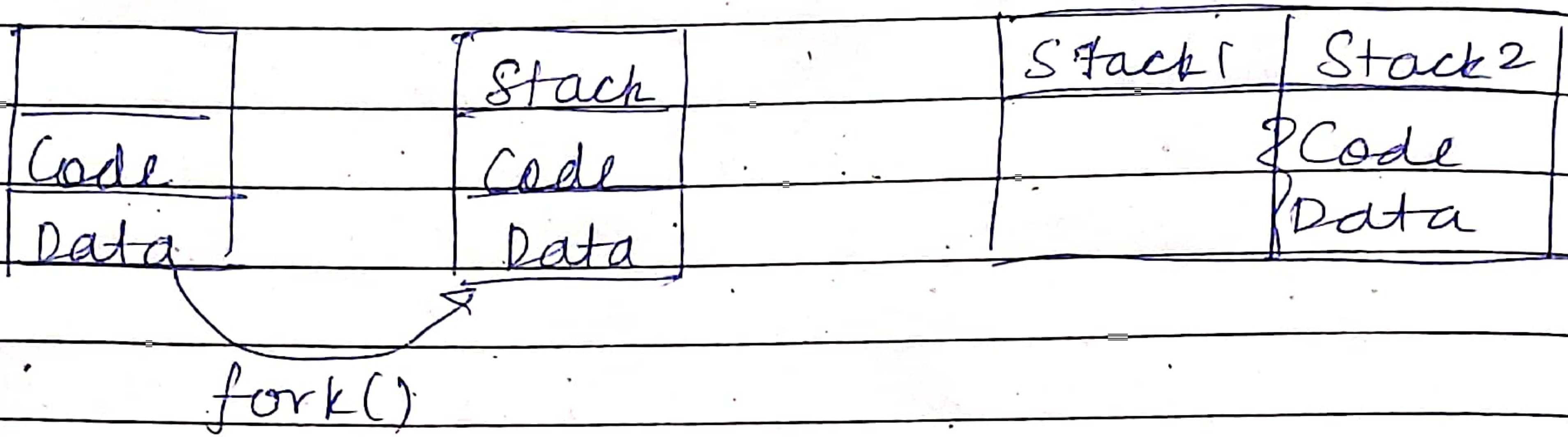
1 ⟵ 0

Process Vs Threads in OS

Process - It is a heavy weight task } Multi-tasking environment

Thread - It is a lightweight task

Process	Thread (User level)
1) System calls involved	1) No system call involved
2) OS treats diff. processes differently	2) All user level threads treated as single task for OS
3) Diff. processes have diff. copies of data files etc..	3) Threads share same copy of code
4) Context switching is slow	4) Context ~~swit~~ switching is fast
5) Blocking a process not block other	5) Blocking a thread block entire process
6) Independent	6) Interdependent


Code
Data
Stack
Code
Data
fork()
Stack1
Stack2
Code
Data

User Level Thread	Kernel Level Thread
1) Managed by user level library	1) Managed by OS (system calls)
2) Typically fast	2) Slower than ULT
3) Context switching is faster	3) Context switching is slower
4) If one ULT performs blocking operation then entire program gets blocked	4) If one KLT is blocked it doesn't affect others.

Context Switch : Process > KLT > ULT