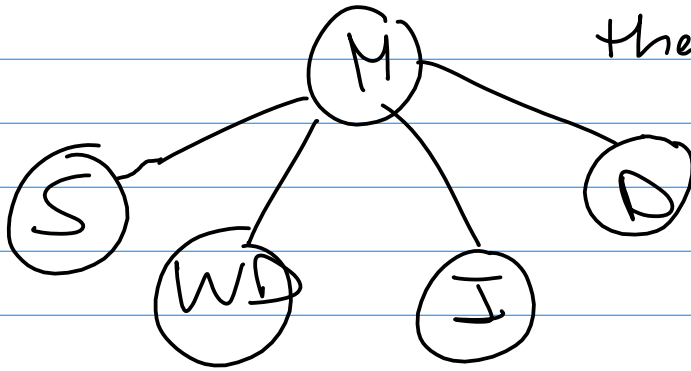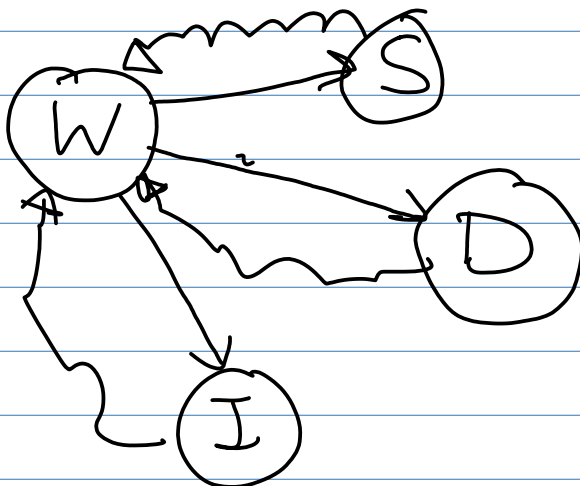server, drone, input, WD

Master : forks and executes all
processes. Before of this,
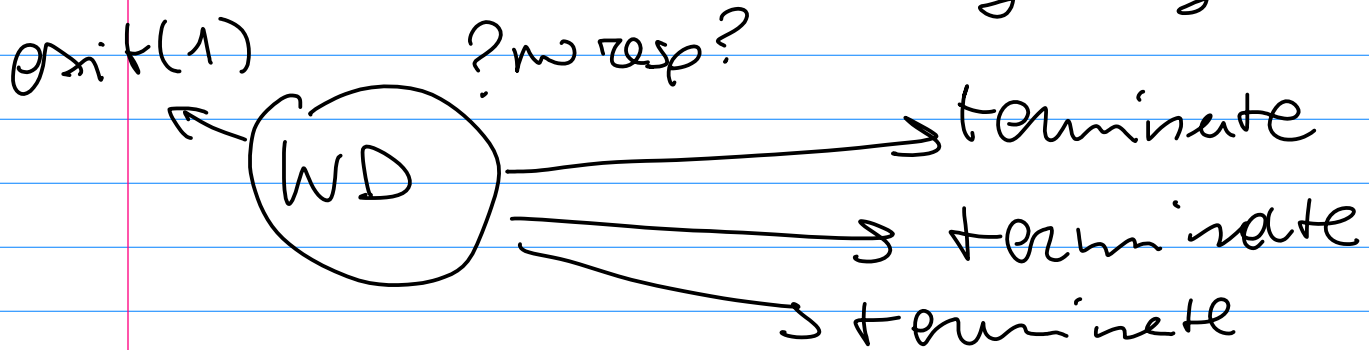there is a descripti
that introduces the
program.

Sends to watchdog the PID
of S, I, D

WATCH DOG (W)

WD sends signals
to all the other
processes

If the WD doesn't receive a
response from one process so
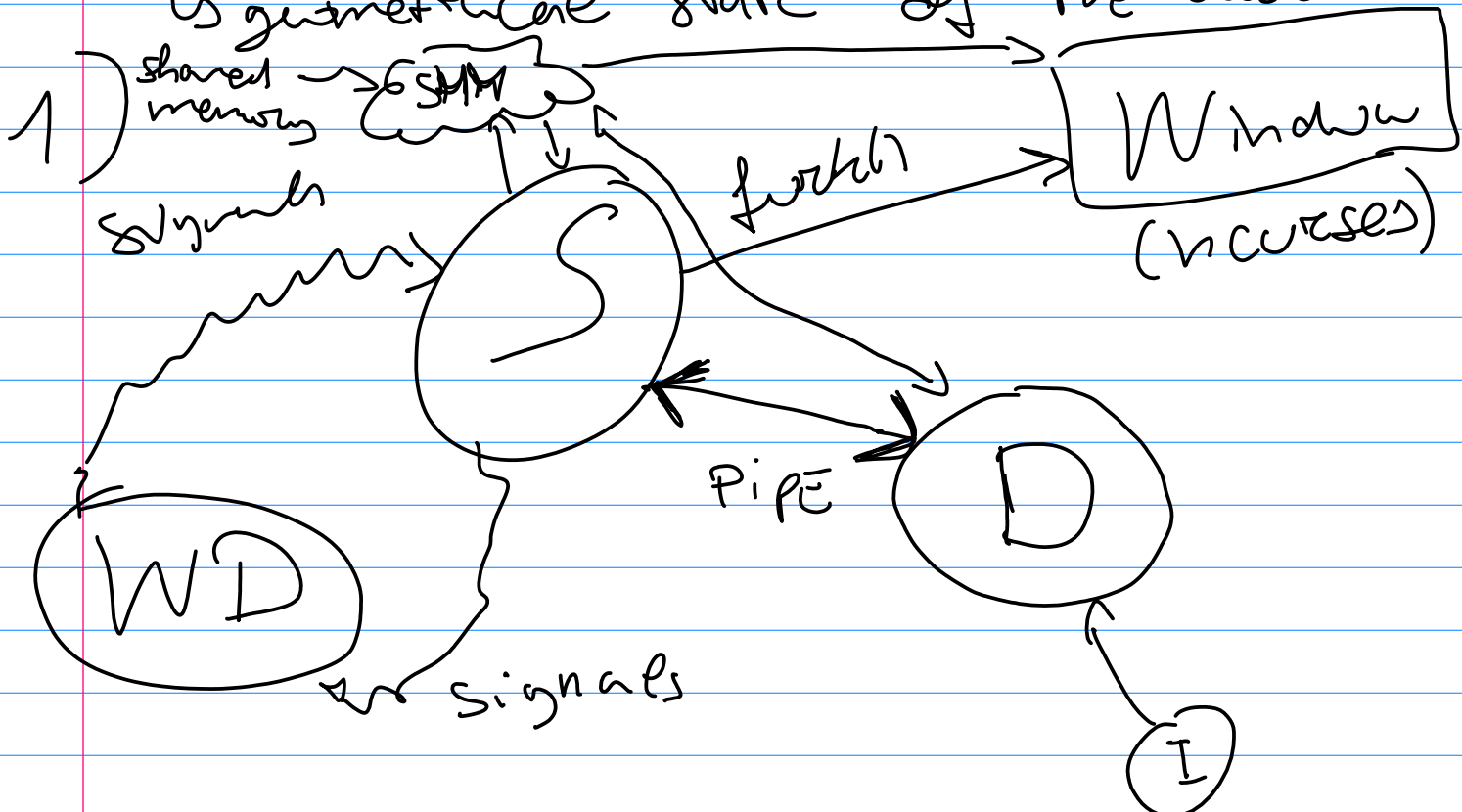it means that there is an anomaly,
it will close everything.

exit(1)   ?no resp?

WD → terminate

→ terminate

→ terminate

SERVER

Uses shared memory   WHICH
                            ARCHITECTURE?

→ Messages BLACKBOARD
  ⟶ geometrical state of the world

1) shared memory → SHM

signals

S → fork()

WD

PIPE

D

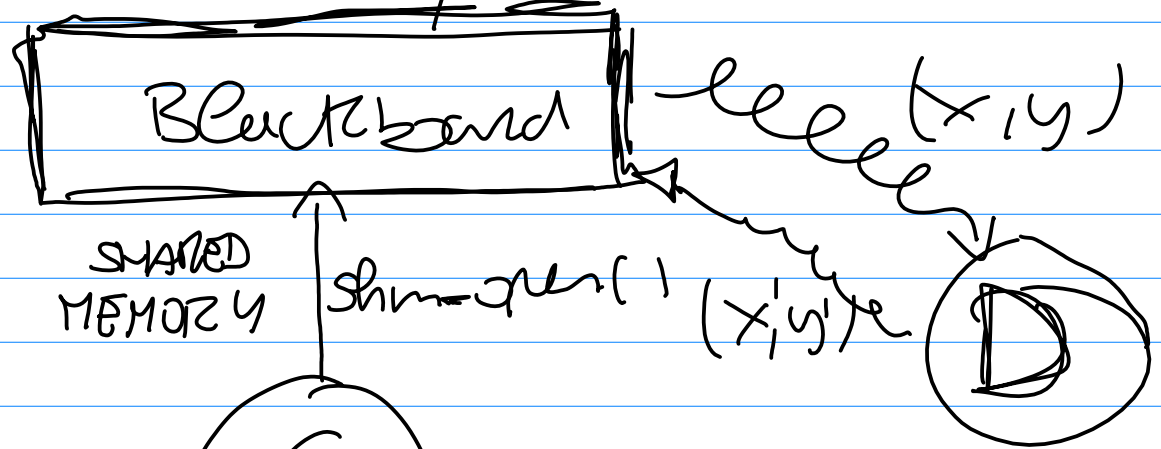signals

Window
(ncurses)

I

- SIMPLER CODE
- LESS LATENCY

2)
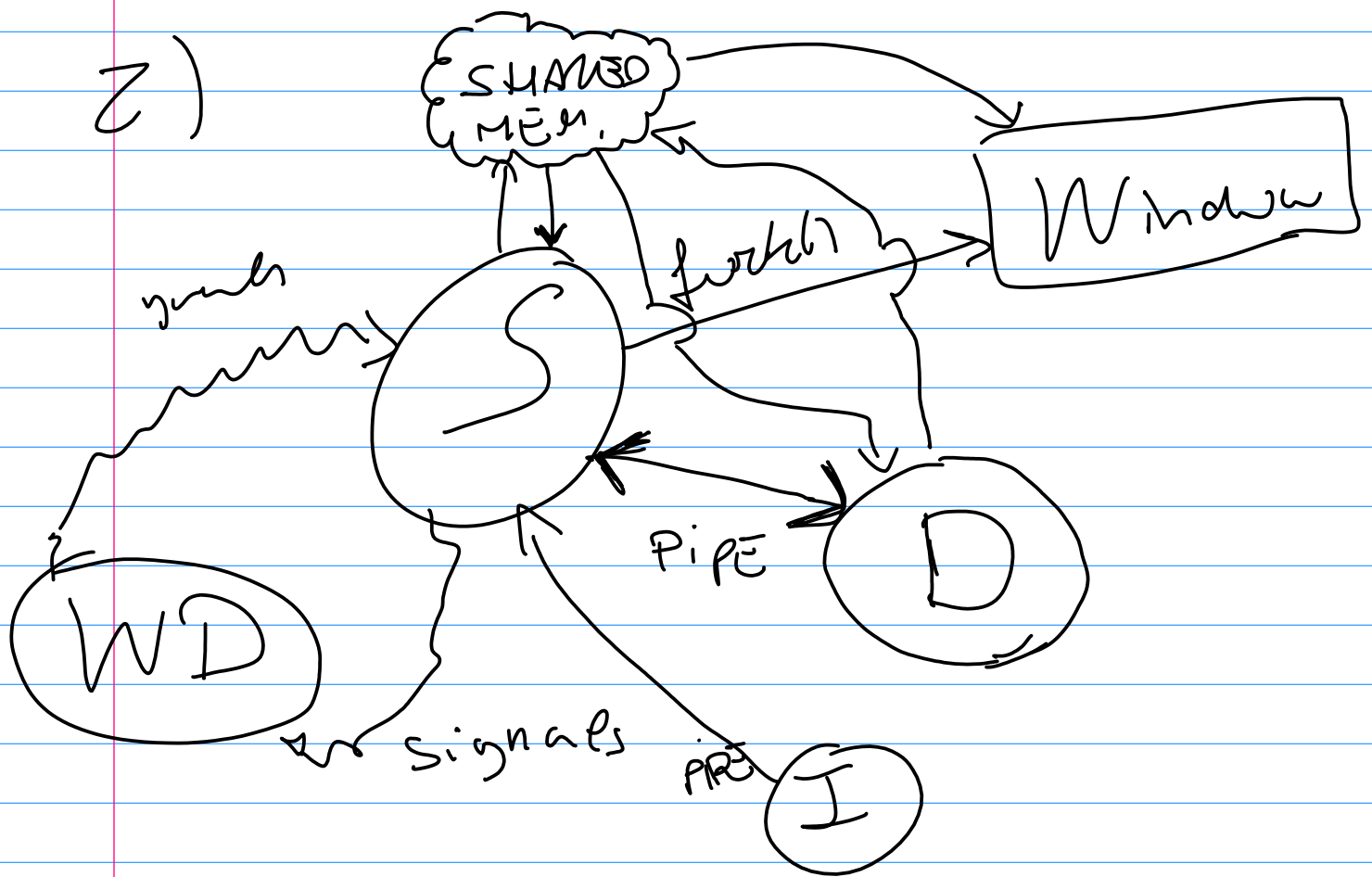
- MORE STABILITY
- BETTER IF I WILL ~~RELEASE~~
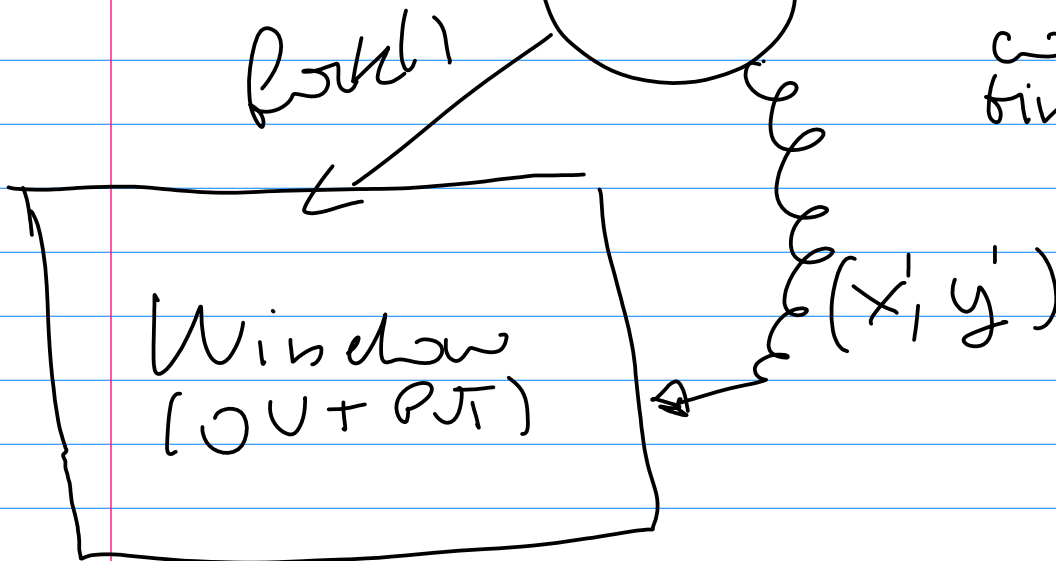UPGRADE THIS SYSTEM

I choose the 1) Architecture

why? → Because in 1st Assignment, the structure is relatively simple to manage.

Let's see better how server
works:

INITIALIZES X, Y of drone

Blackboard $\longrightarrow$ (x, y)

SHARED
MEMORY $shm\_open()$ (x', y') D

x', y' are
coordinates at
time $t+1$

$fork()$

S

Window
(GUI + PJT) (x', y')

DRUG

$T = $ periodo

$m \to$ massa $\omega \div$ 2oms

$k \to$ coeff. di viscosità

$p \to$ posizione

$$F_{TOT} = m\frac{d^2}{dt^2}P + k\frac{dp}{dt}$$

$\downarrow$

$$F_{TOT} = m \cdot a + k \, v$$

By pressing a key you increment of mod the force in the direction selected.

To this force, there is the friction force that contrasts the motion:

generated $\to F - k \cdot v = m \cdot a$ $\left(\begin{array}{l}\text{2nd Newton's} \\ \text{Law}\end{array}\right)$

attrito

SO, calling V', X', y' the coordinates and velocity at time $t+T$:

$$a = \frac{F - Kv}{m}$$

$\downarrow$ integrate in $t \to t+T$

$$V' = V + a \cdot T$$

$\downarrow$ " " " "

$$\begin{cases} x' = x + V'_x T + \frac{a}{2} T^2 \\ y' = y + V'_y T + \frac{a}{2} T^2 \end{cases}$$

INPUT

Reads characters and then sends them with a pipe to drone.

character

INPUT → character pipe() → DRAW

pipe()

GRAPHIC
INTERFACE
OF INPUT
(ncurses)