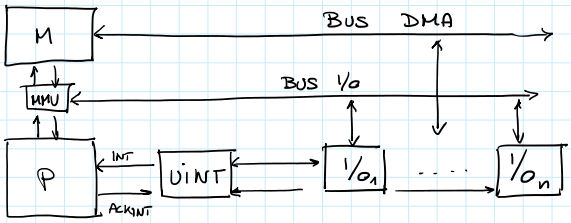
Ingresso/uscita & interruzioni

lunedì 27 novembre 2017 09:20



add1 ... (INT = 0) ..., chs
add0 ... = 1) in tratt_int

tratt_int. reset(INT), set ACKINT, int1

int1 . (RDYinM, OR(ESITO) = Ø-) map, int1
(=10) reset RDYinM, set ACKM, DATAIN → REG[61], int2
(=11) reset RDYinM, tratt_ecc

int2 . (RDYinM, OR(ESITO) = 0 -) map, int2
(=10) reset RDYinM, set ACKM, DATAIN → REG[62],
IC → REG[63], REG[60] → IC, chs
(=11) reset RDYinM, tratt_ecc

fase firmware

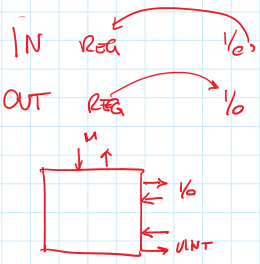
LOAD Rtab_int, R61, Rhandler, DI
CALL Rhandler, Rret
GOTO R63, EI

fase assembler

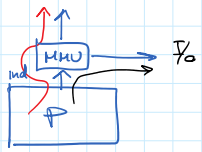
VETT INT

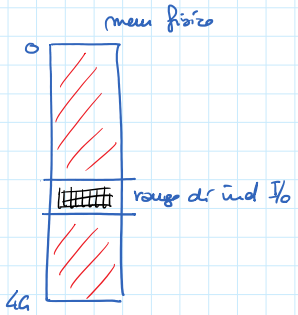
i IND DRIVER della PERIF;

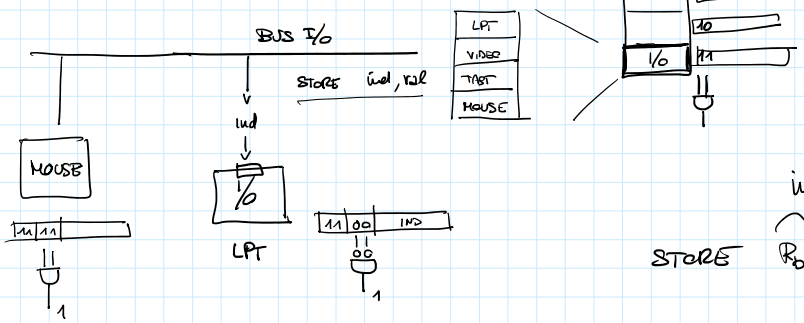
I/O con istruzioni speciali



Memory mapped I/O







ind ∈ I/O

STORE R0, Ri, R1

LATO PERIFERICA (unità di I/O)

sse mi riconosco come destinatario di uno STORE IND, VAL

a) ind come indirizzo di una mia cella interna

b) interpreto <ind, val> direttamente a livello FW

per esempio val → comando da eseguire

ordine lettura del settore X (os)

STORE $R_{base\text{-}periferica\,disco}$, #1, R_X

STORE $R_{base\text{-}periferica\,disco}$, #0, $R_{"READ"}$

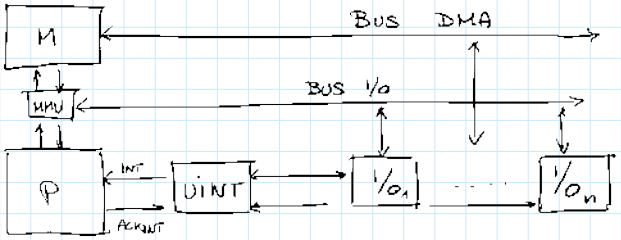
↑

lettura dei parametri dalla $M_{I/O} \rightarrow M$ (leggo e.g. coord. mouse)

LOAD $R_{base\text{-}perif\_mouse}$, #1, R_{dx}

LOAD $R_{base\text{-}perif\_mouse}$, #2, R_{dy}

DMA direct memory access (I/O)



a) I/O ha uno proprio MMU e usa indirizzi che vengono tradotti internamente in fisici per accedere alla M

b) I/O non ha uno MMU e usa direttamente indirizzi fisici

Operazioni "a blocchi"

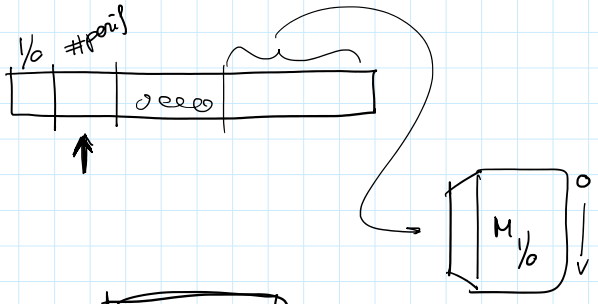
1) nei param dell'op che viene ordinato all'I/O ci sono gli indirizzi dei buffer in memoria
2) I/O esegue il ciclo di trasferimento

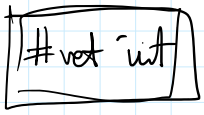
READ (blocco (x), md (Y)) ⟶
- x → disco
- Y → in M

STORE	X
STORE	Y
STORE	N
STORE	"READ

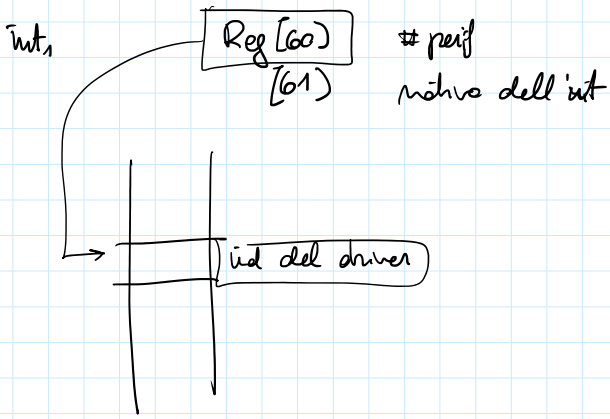
⟶ WAIT ⇒ cambio contesto

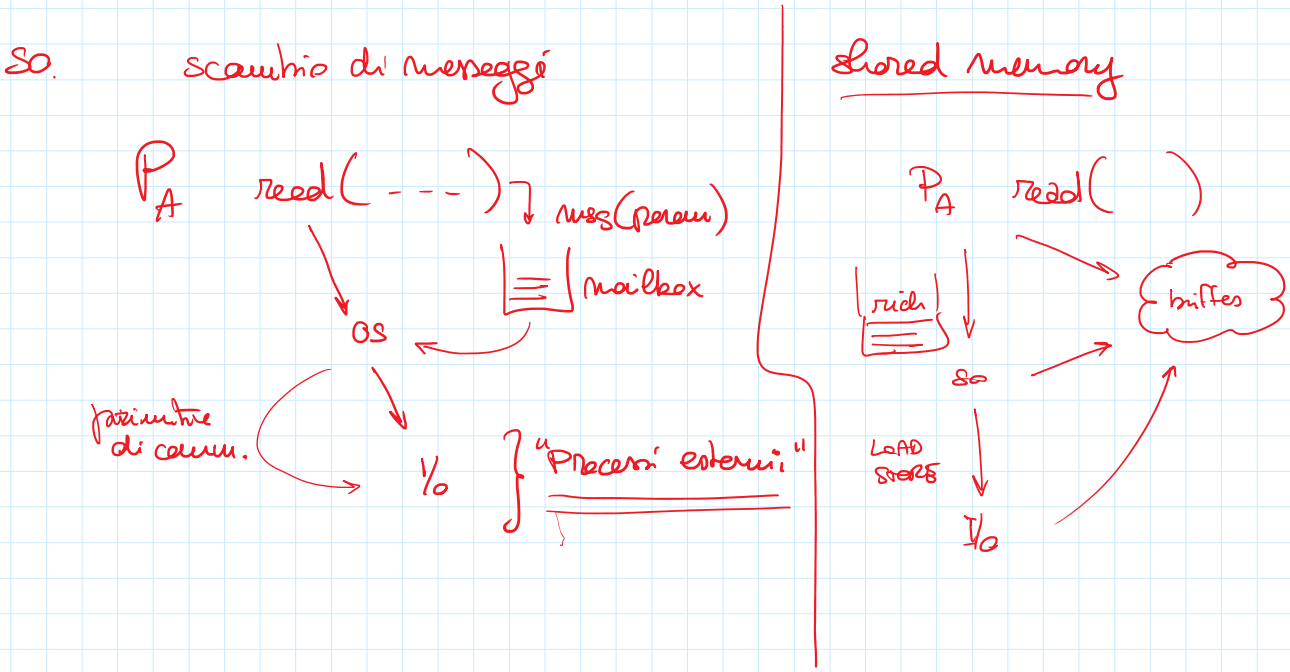
⟶ INT di fine lettura
se l'op è andato a buon fine
i dati saranno in M all'indirizzo Y


I/O
#perif
M
I/O


#vet int

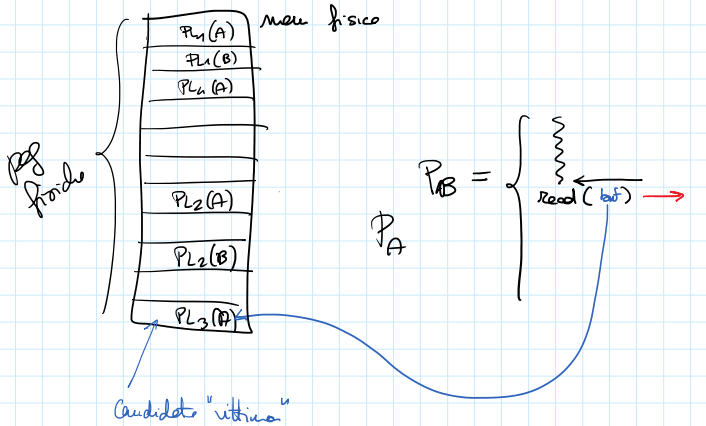
plug & play


int,
Reg [60]
[61]
perif
motivo dell'int
ind del driver


SO
scambio di messaggi
shared memory
P_A
read(- - -)
mess(param)
mailbox
OS
primitive di comun.
I/O
"Processi esterni"
P_A
read( )
richi
buffer
SO
LOAD
STORE
I/O

lunedì 27 novembre 2017 10:51

DMA + gestione della memoria (paginazione dinamica)



Candidato "vittima"
× la paginazione dinamica
da sostituire $PL_x(C)$

quando avviene l'accesso DMA
M I/O → M : dati letti

a) P_B è sicuramente in WAIT

per evitare problemi
$PL_3(A)$ deve essere
marcata come "non deallocare"
(non swappabile)