

Attenzione: questa non è una politica a priorità classica.

Si tratta di una versione speciale (potremmo chiamarla *fair priority*, ovviamente con prerilascio)

Ogni X unità-tempo consecutive in esecuzione:

Priorità := - floor(X),

Priorità ≥ 0 ,

Una volta abbassata la priorità, essa non viene più rialzata.

La priorità non può scendere sotto lo 0

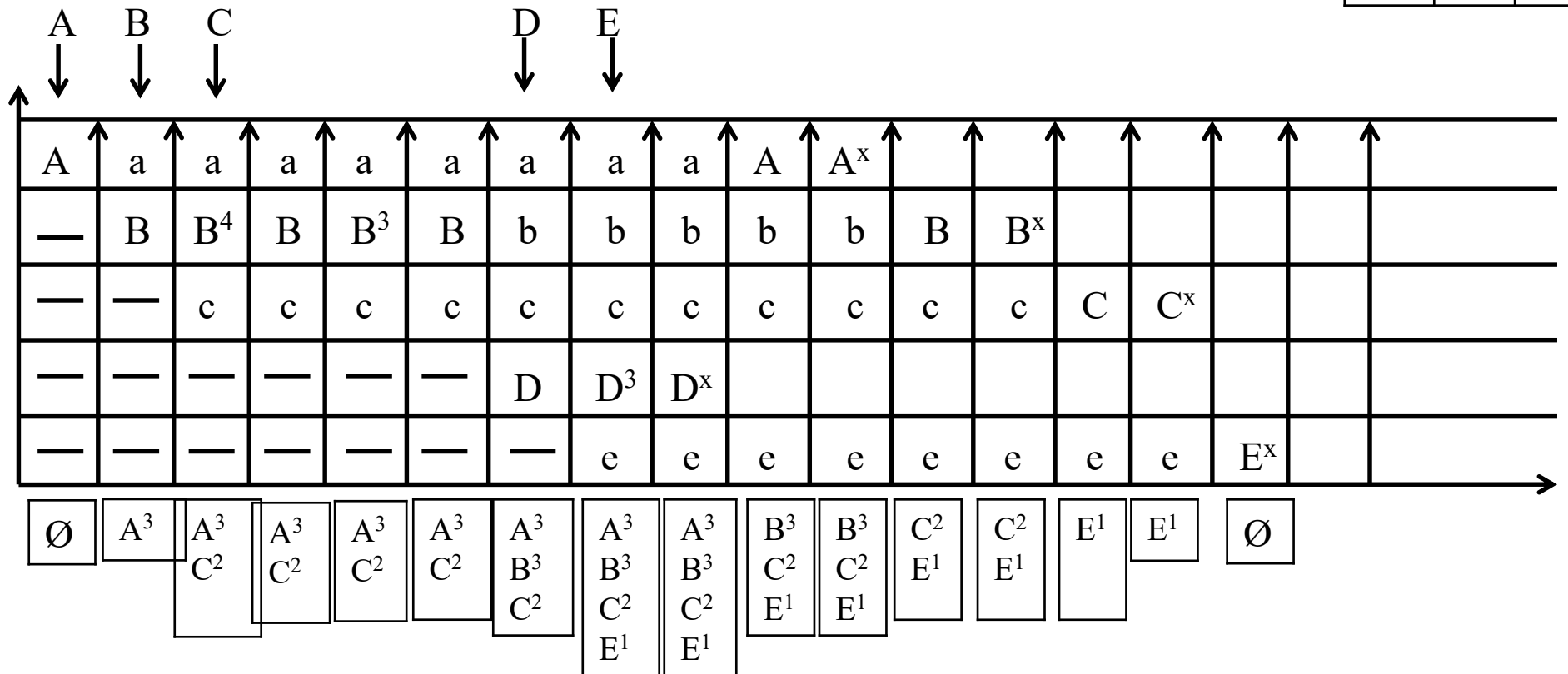
“Nel caso di arrivo di un processo in contemporanea a un’uscita per `time_out()`, si dia precedenza al processo prerilasciato per `time_out()`”

(Applicata per eventi uscita/arrivo appartenenti alla stessa classe priorità)

Pid	T_arr	T_exec	Priority
A	0	3	3
B	1	7	5
C	2	2	2
D	6	3	4
E	7	1	1

**Fair Priority con $X = 2$
 $\text{floor}(X) = 1$**

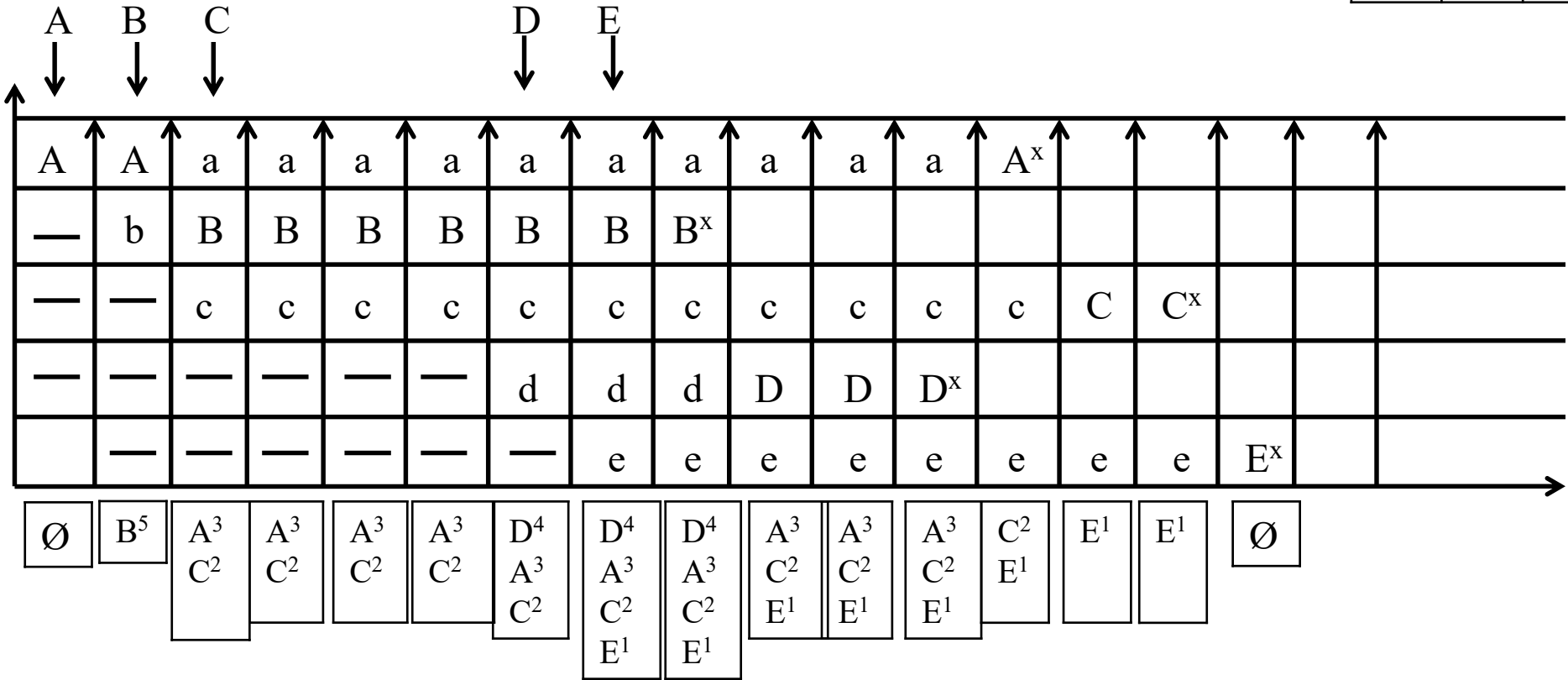
T_att	T_ta	T_r
8	11	0
5	12	0
11	13	11
0	3	0
8	9	8
6.4	9.6	3.8



Pid	T_arr	T_exec	Priority
A	0	3	3
B	1	7	5
C	2	2	2
D	6	3	4
E	7	1	1

RR con priorità senza prerilascio,
 Quanto = 2 [u.t]

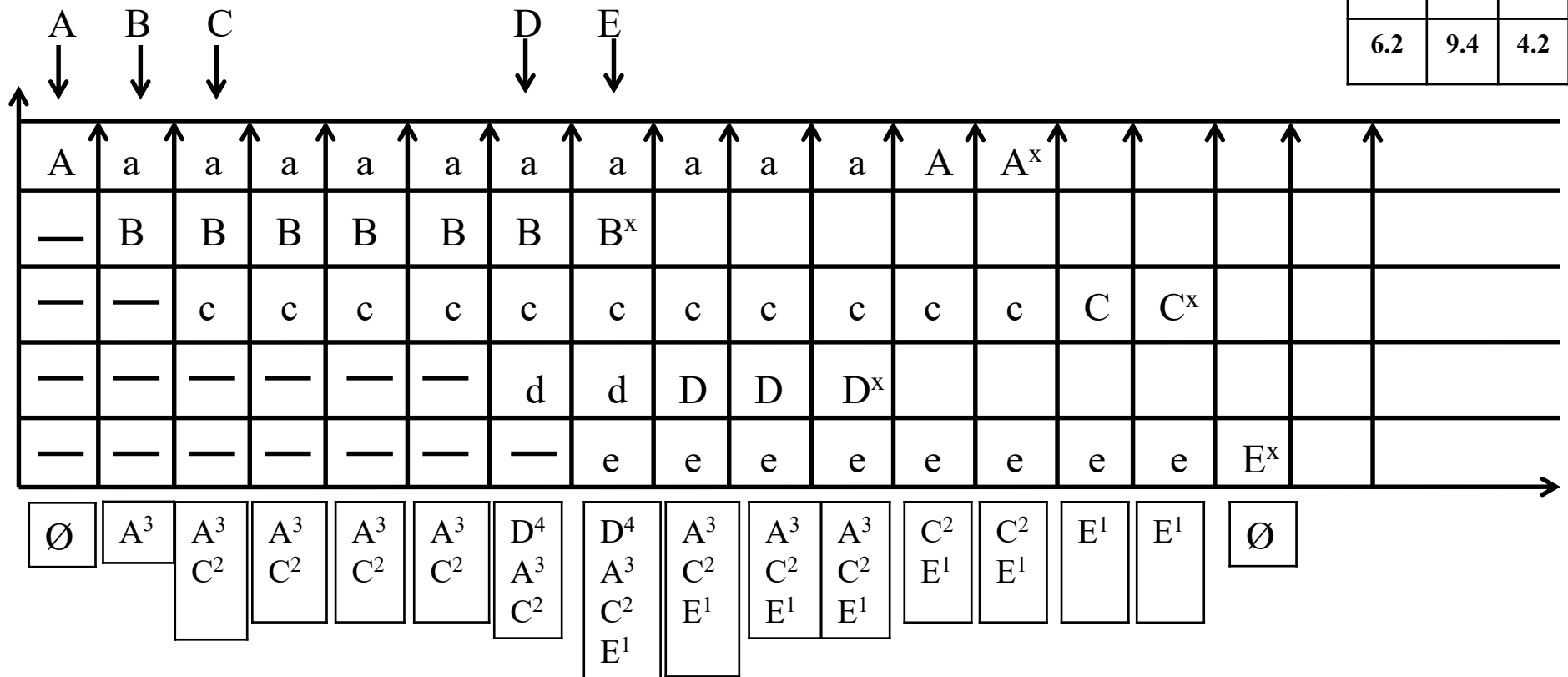
T_att	T_ta	T_r
10	13	0
1	8	1
11	13	11
3	6	3
8	9	8
6.6	9.8	4.6



Pid	T_arr	T_exec	Priority
A	0	3	3
B	1	7	5
C	2	2	2
D	6	3	4
E	7	1	1

RR con priorità con prerilascio,
 Quanto = 2 [u.t]

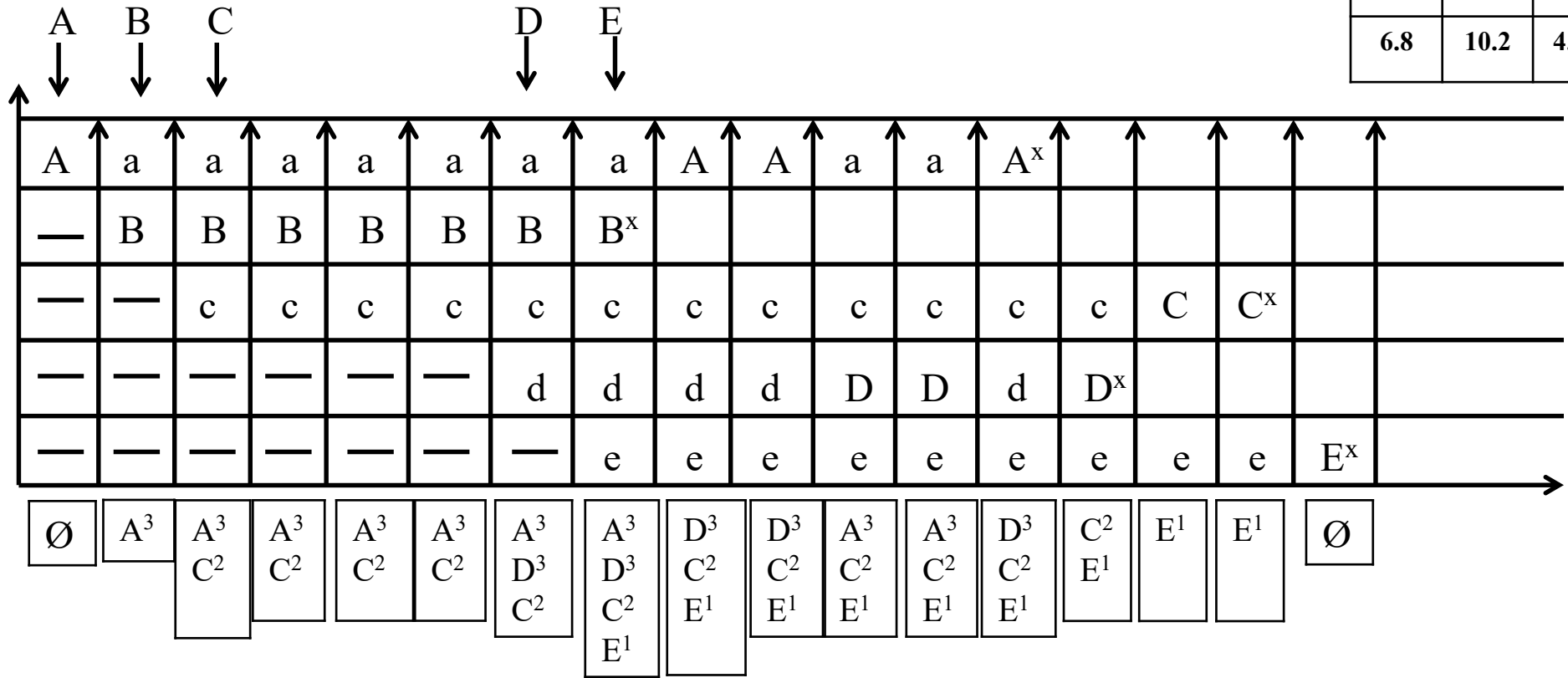
T_att	T_ta	T_r
10	13	0
0	7	0
11	13	11
2	5	2
8	9	8
6.2	9.4	4.2



Pid	T_arr	T_exec	Priority
A	0	4	3
B	1	7	5
C	2	2	2
D	6	3	3
E	7	1	1

RR con priorità con prerilascio,
 Quanto = 2 [u.t]

T_att	T_ta	T_r
9	13	0
0	7	0
11	13	11
5	8	4
9	10	9
6.8	10.2	4.8



Pid	T_arr	T_exec	Priority
A	0	3	3
B	1	7	5
C	2	2	2
D	6	3	4
E	7	1	1

SJF con prerilascio (ovvero **SRTN**)

T_att	T_ta	T_r
0	3	0
8	15	4
1	3	1
1	4	0
0	1	0
2	5.2	1

