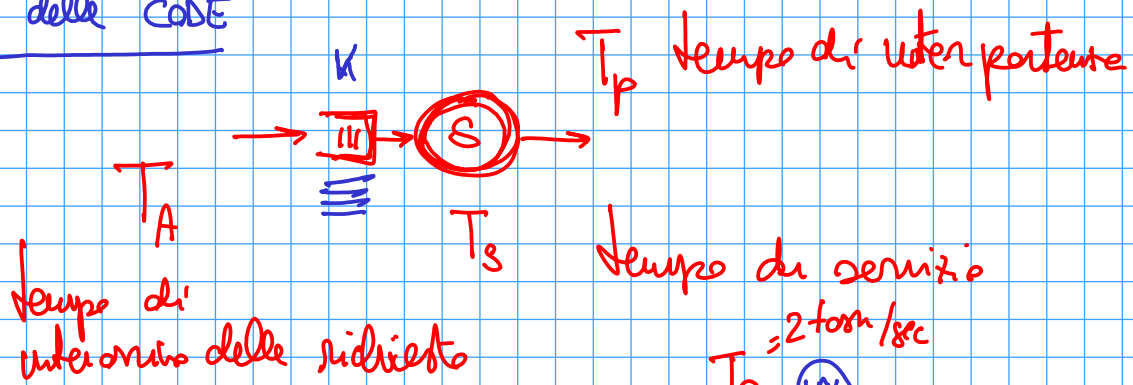
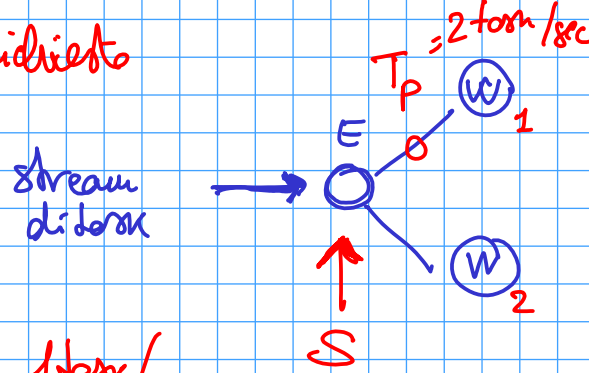


# TEORIA delle CODE

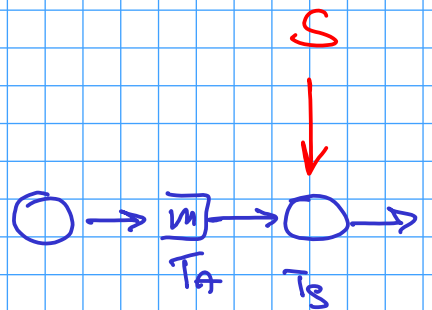


lo code def il grado di occupazione



$$\rho = \frac{T_s}{T_A}$$

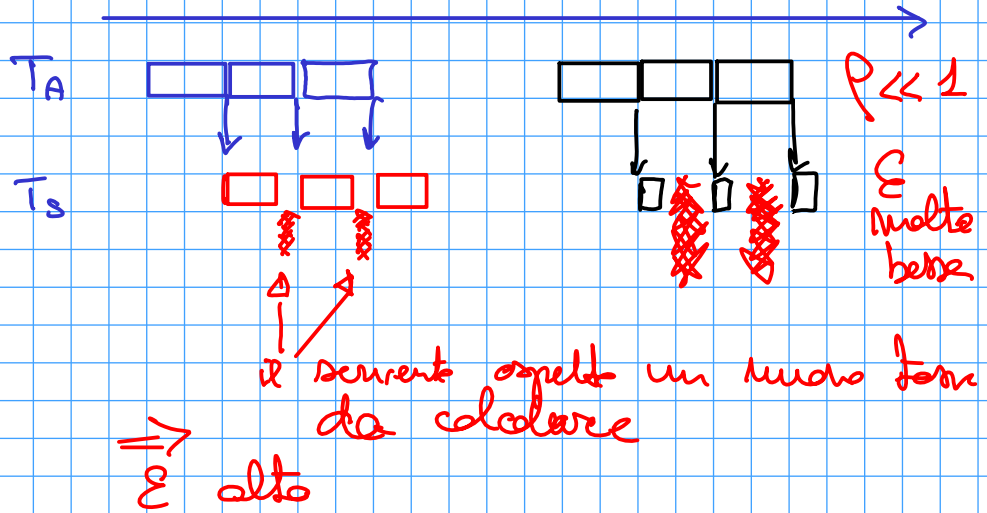
$\rho \geq 1$  "lo code si riempie"  
 $\rho < 1$  "la code funziona"



rho fattore di utilizzo delle code

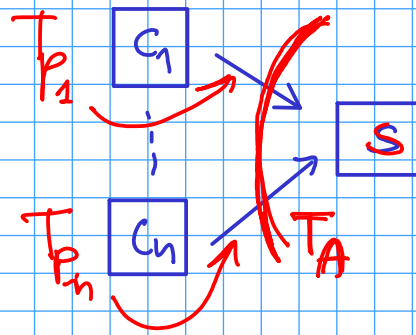
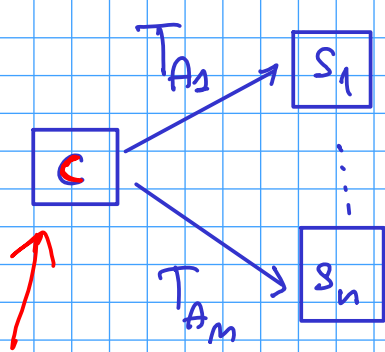
$$\rho = \frac{T_s}{T_A} \approx 1 \text{ per il minore}$$

$\rho < 1$



code limitate (k - posizioni)

+ code piena  $\Rightarrow$  allocazione "di code di trabocco"  
 minore code limitate (k)



$$T_A = \frac{1}{\sum_{i=1}^n \frac{1}{T_{P_i}}}$$

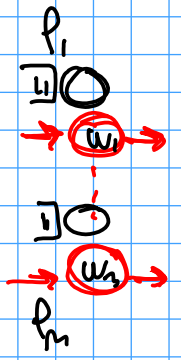
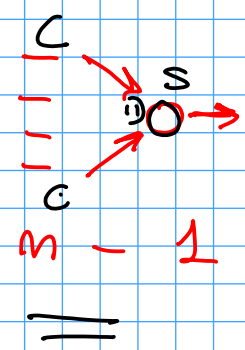
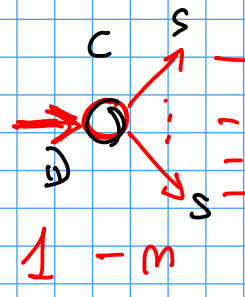
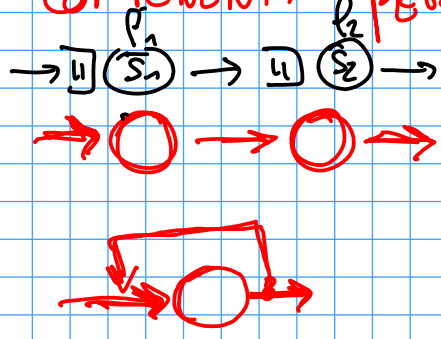
Conoscenza dei  
parametri di c

$T_P$   $P_i$  prob. che un box  
sia diretto al  
server  $S_i$   
tempo  
di interpretazione

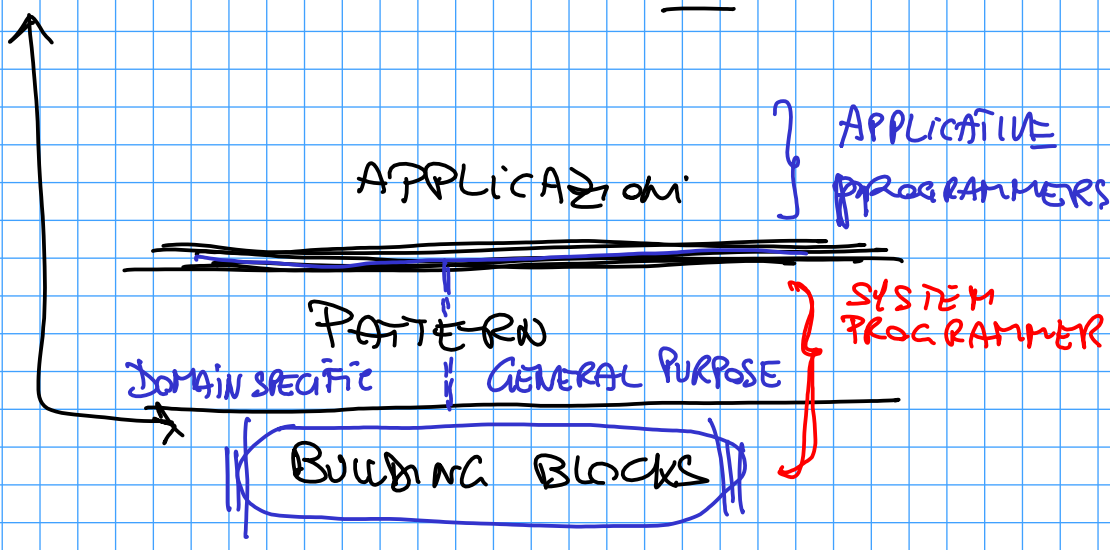
$$T_{A_i} = \frac{T_P}{P_i} = \frac{1 \text{ box/sec}}{\frac{1}{2}} = 2 \text{ box/sec}$$

$$T_{P_c} \quad T_{S_i} \approx T_{P_c} \Rightarrow E \text{ molto bene}$$

"COMPONENTS PER COSTRUIRE PATTERN"

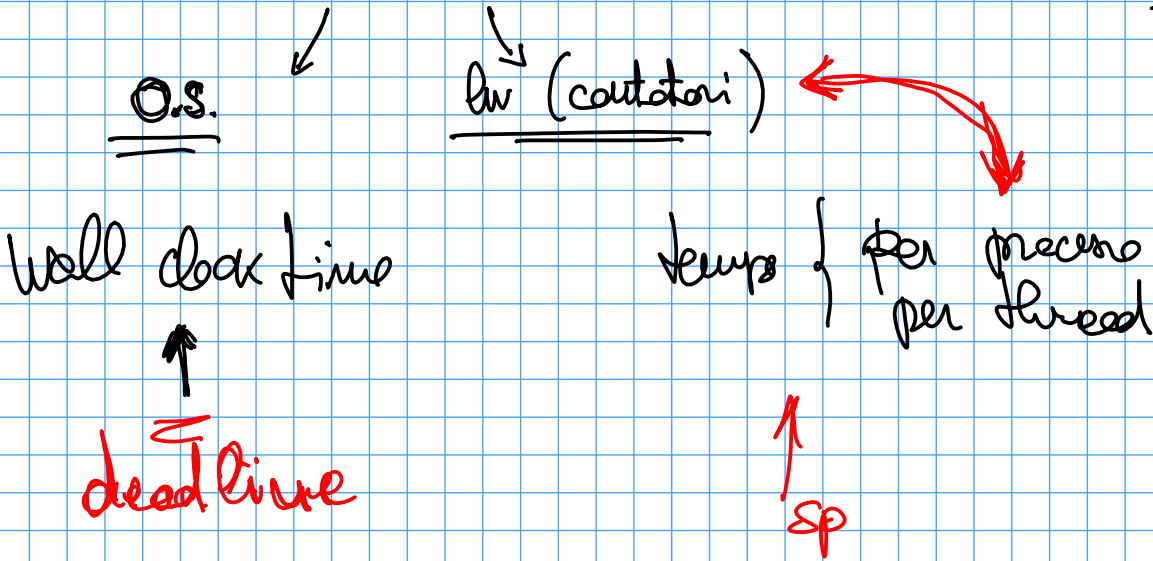


~~RISC~~



Posix ⇒ Linux (MAC OS X) ((Windows))

meccanismi x misurare il tempo (speso in una parte di codice)



comando de shell > time cmd param<sub>1</sub> ... param<sub>n</sub>  
real  
user  
system

chiamate di libreria

gettimeofday (

Codice

) gettimeofday (

t<sub>0</sub>

t<sub>1</sub>

struct timeval {  
tv\_sec;  
tv\_usec;

t<sub>1</sub>.tv\_usec -  
t<sub>0</sub>.tv\_usec

```
struct timeval t0, t1;  
gettimeofday(&t0, NULL);  
...  
gettimeofday(&t1, NULL);
```

clock - get time

