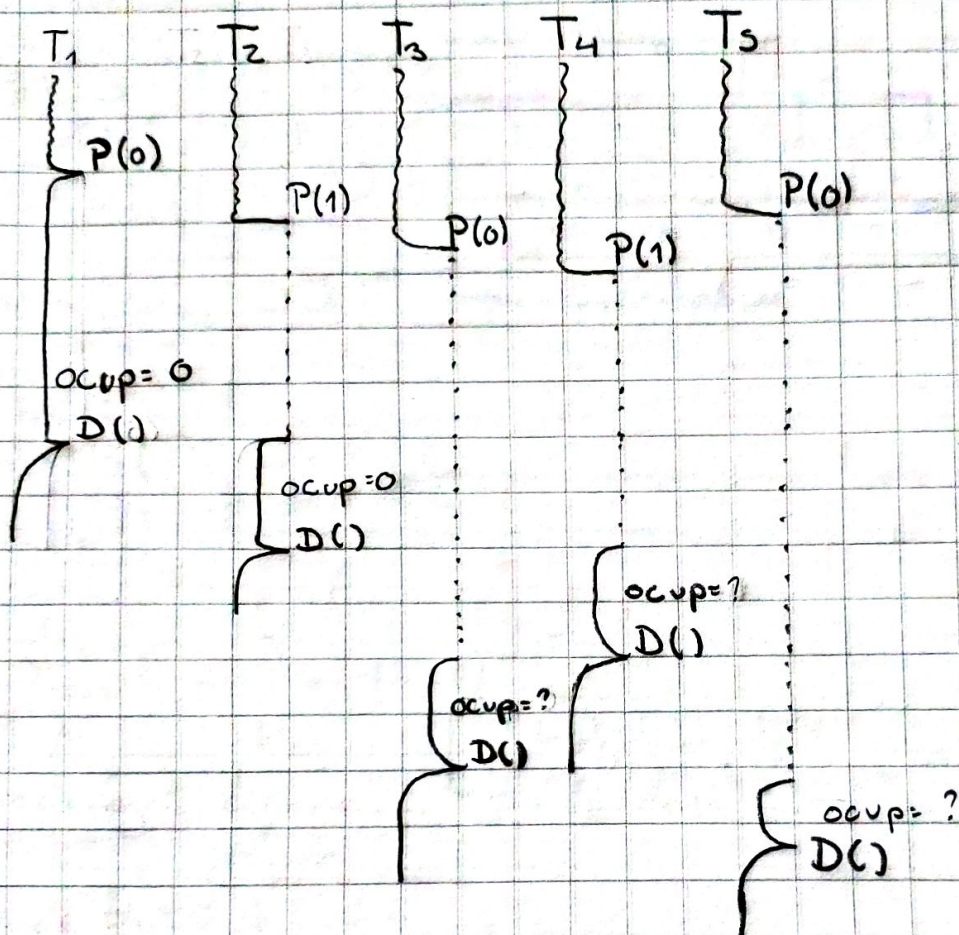


C1 S.O.

Diego Hurtado

Pregunta 1:

a)



- El problema que se presenta es que se pierde el control sobre la variable `ocup`, ya que su valor se reasigna después de ceder el ticket.
- Se puede dar el curso en que `ocup` sea 0 mientras el proceso que está trabajando tenga `cart = 1`.

P1 b y P1 c están adjuntados como txt.



P2]

a) `int ocup;`  
`Queue cats1, cats0;`  
`void init() {`  
    `ocup = -1;`  
    `cats0 = MakeQueue();`  
    `cats1 = MakeQueue();`  
`}`

• A la estructura `nTask` le agrego un campo `int num;` para guardar su valor `cat`

`void nPedir(int cat) {`  
    `START_CRITICAL();`  
    `nTask tarea = current-task;`  
    `tarea->num = cat;`  
  
    `if (ocup == -1) {`  
        `tarea->status = READY;`  
        `ocup = cat; PushTask(ready-queue, tarea);`  
    }  
    `else {`  
        `tarea->status = Wait-Pedir;`  
        `if (cat == 0) {`  
            `PutTask(cats0, tarea);`  
        }  
        `else {`  
            `PutTask(cats1, tarea);`  
        }  
    }  
    `ResumeNextReadyTask();`  
    `END_CRITICAL();`  
`}`



```

void nDevolver() {
    START_CRITICAL();
    if (ocup == 0 && !(EmptyQueue(cats 1))) {
        nTask act = GetTask(cats 1);
        act -> status = READY;
        ocup = act -> num;
        ResumeNextReadyTask();
    } else if (ocup == 1 && !(EmptyQueue(cats 0))) {
        nTask act = GetTask(cats 0);
        act -> status = READY;
        ocup = act -> num;
        ResumeNextReadyTask();
    } else if (ocup == 0 && !(EmptyQueue(cats 0))) {
        nTask act = GetTask(cats 0);
        act -> status = READY;
        ocup = act -> num;
        ResumeNextReadyTask();
    } else if (ocup == 1 && !(EmptyQueue(cats 1))) {
        nTask act = GetTask(cats 1);
        act -> status = READY;
        ocup = act -> num;
        ResumeNextReadyTask();
    } else {
        ocup = -1;
    }
    END_CRITICAL();
}

```

\* Perdon lo desordenado y duplicado, mientras lo escribia me di cuenta y no tengo refactor :)

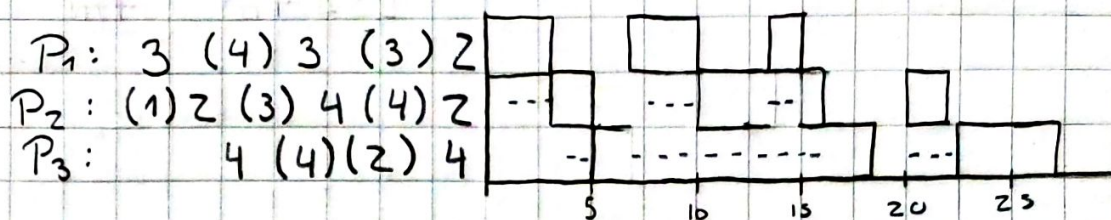


Cn S.O.

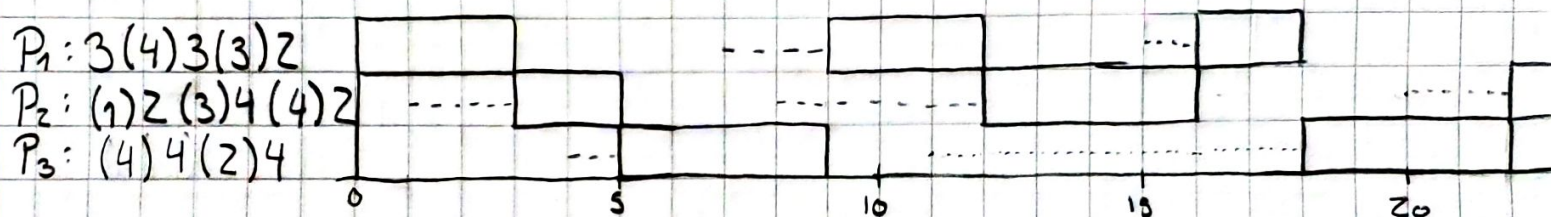
Diego Huilcoles

P2]

b-...



Ahora con SJF non-preemptive



formulas:

$$T_{n+1}^P = \alpha t_n^P + (1-\alpha) T_n^P$$

Pred. prox rafaga

duracion  
rafaga

Predictor de la ejecutado  
ponderador

Tomamos  $\alpha = 0,5$

$$T_{n+1}^P = \alpha t_n^P + (1-\alpha) \alpha t_{n-1}^P + (1-\alpha)^2 \alpha t_{n-2}^P \dots$$

$$T_3 = \alpha \cdot 2 + (1-\alpha) \alpha \cdot 3$$

LEER



El total demora 24, el dibujo no lo me llega hasta el 23 pero ese era el ultimo y duraba 2.

→ Como es non-preemptive la formula solo sirve para ver cual toca pero lo rafaga lo termina completa