

**Control de Laboratori**  
**Grup 30**  
**Contesteu al mateix full**

---

Nom i Cognoms:

---

1. (3 punts) Seleccioneu la resposta ( una només ) que considereu correcta en cadascún dels apartats. Cada resposta correcta val 0,5 punts. Cada resposta incorrecta resta 0,25.
- (a) Indica quina de les següents afirmacions es certa:
- La rutina ksignal pot cridar-se sobre qual-sevol semàfor del sistema.
  - El gestor de teclat ha d'utilitzar el primer semàfor de kernel.
  - Els processos d'usuari no poden utilitzar els semàfors que implementa el sistema.
- (b) Quan es destrueix un procés bloquejat a un semàfor
- S'ha d'alliberar el seu pcb i la seva pila.
  - S'ha d'alliberar el seu pcb, la seva pila i el semàfor si era l'últim procés bloquejat.
  - S'ha d'alliberar el seu pcb i la seva pila. El procés s'ha d'extreure del semàfor.
- (c) L'estructura info\_proc
- S'ha de definir a nucli.h.
  - S'ha de definir a onionlib.h.
  - S'ha de definir a nucli.h i onionlib.h.
- (d) La rutina planificacio\_wakeup s'ha de cridar
- sempre que vulguem posar un procés en execució.
  - sempre que un procés es desbloqueja.
  - sempre que un procés entra a la cua de ready.
- (e) El codi del gestor de teclat s'ha d'inhibir perquè:
- ha d'accedir a les cues en exclusió mutua.
  - el codi de sistema s'executa sempre inhibit
  - la entrada/sortida es fa inhibida
- (f) Quina de les següents afirmacions es falsa:
- Els semàfors poden fer servir el camp valor per saber si són inicialitzats
  - El gestor de teclat es l'únic que pot cridar llegir\_teclat\_nuc.
  - L'iorb de la petició de teclat es pot declarar a la pila.
2. (1 punts) Explica perquè necessitem el context de kernel de la tercera entrega.

3. (4 punts) Indica a quines rutines creus que pertanyen els següents troços de codi. El nom de la rutina ha de ser significatiu d'ho que fa i d'on es troba situada al nucli.

```

id_io = buscar_sem_lliuere() .....
planificacio_wakeup(bloq_teclat) .....
encuar(&qpiles_lliuere,pcb→pila) .....
ksem_init(sem_gestor,0) .....
inicialitzar_cua(q_bloqs) .....
c = llegir_teclat_bfs() .....
planificacio_wakeup(primer(q_bloqs)) .....
case PCB_RETARD: retardat = 0 .....

```

4. (2 punts) Completa el codi de la rutina `modif_proc_nuc`

```

void modif_proc_nuc ( int pid, struct info *info)
{
    PCB *proces;

    if ( info->prioritat < MIN_PRIORITAT || info->quantum < MIN_QUANTUM ||
        pid < MIN_PID || pid == PID_NUL || (proces = buscar_pcb(pid)) == 0 ) {
        RUN->ctx->ax = -1;
        return;
    }
}

```

```

}

```