

SISTEMA OPERATIU INICIAL

- **Monoprogramat:** Varis programes carregats en memòria però el què s'està executant ho fa fins el final, voluntària o involuntàriament

- Fins a `liberar_proceso` no es crida el planificador i es fa canvi de context.

- `Lista_listos`

- 3 crides ja definides a sistema

VERSIÓ NOVA (PRÀCTICA1)

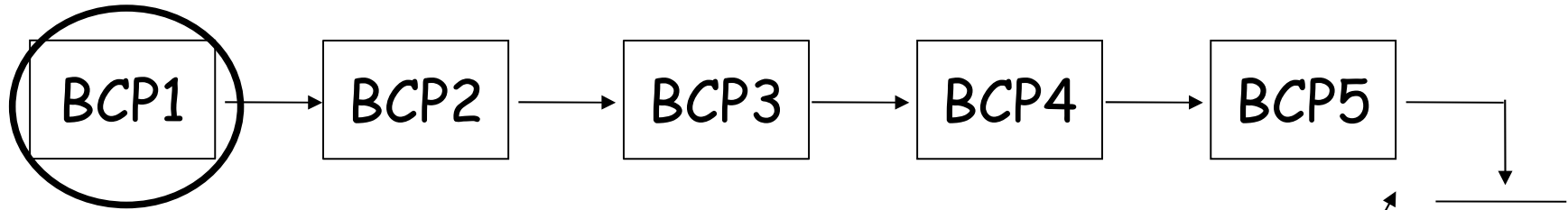
- **Multiprogramat:** Varis programes carregats en memòria però el què s'està executant es pot bloquejar (dormir). Llavors, un altre procés en estat de `LISTO` pot passar a ocupar el processador.

- En bloquejar el procés, es crida el planificador i es fa un canvi de context voluntari.
- Temporitzador per a despertar els processos dormits.

- `Lista_listos` i `lista_dormidos`

- Nova crida disponible en el kernel: `sis_dormir`

lista_listos (kernel.c)



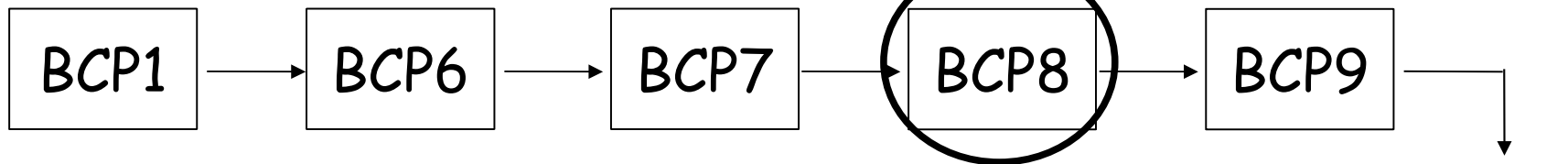
sis_dormir (kernel.c)

```
obtener_segs(Leer_registro(1))
calcular_tics
bloquear(lista_dormidos)
```

int_reloj() (kernel.c)

```
per tot proc e lista_dormidos
tics--
si tics == 0
    desbloquear(lista_dormidos,
                proc)
```

ajustar_dormidos



lista_dormidos (kernel.c)

Tractament de les llistes

- Mètodes de manegament de les llistes:
 - insertar_ultimo (lista_BCPs *lista, BCP * proc)
 - eliminar_primer (lista_BCPs *lista)
 - eliminar_elem (lista_BCPs *lista, BCP * proc)
- Important!!: Mai no pot haver alhora un mateix BCP a dues llistes:
 - Primer s'ha d'eliminar un BCP de la llista
 - Després s'ha d'inserir a l'altre.

bloquear (lista_BCP *lista)

- Rutina que passa el procés actual a bloquejat.
- El procés actual es treu de la llista lista_listos
- La llista de BCP's que es passa com a paràmetre és la llista on es bloquejarà el procés
- És necessari identificar un nou procés actual i realitzar el pertinent canvi de context
- Detall: s'ha de controlar una possible interrupció de nivell de prioritat superior:
 - definir la prioritat de quines interrupcions s'han d'inhibir
 - definir el tros de codi mínim on és necessari inhibir aquestes interrupcions