

Examen la Sisteme de operare 2

- probleme -

18.05.2004

1. [2.5 puncte] Să se implementeze apelurile sistem necesare unui mecanism simplu de comunicare între procese prin semafoare:

- `int sem_init(int value);`
Crează un nou semafor și îl inițializează cu valoarea `value`.
- `int sem_down(int s, int value);`
Operația *down* pe semaforul `s`, cu valoarea `value`.
- `int sem_up(int s, int value);`
Operația *up* pe semaforul `s`, cu valoarea `value`.
- `int sem_get(int s, int *value);`
Depune valoarea semaforului `s` în `value`.

Toate funcțiile întorc `-1` în cazul apariției unei erori. Funcția `sem_init` întoarce un *identificator* pozitiv care desemnează semaforul utilizat în celelalte operații. Restul apelurilor sistem vor returna `0` dacă operația s-a efectuat cu succes.

Apelurile sistem `sem_up` și `sem_down` incrementează/decrementează valoarea semaforului (atunci când este posibil) cu valoarea `value`.

Toate operațiile de mai sus vor fi atomice (mai puțin, eventual, `sem_init`).

Se poate considera că există un număr maxim oarecare posibil de semafoare.

Macro-ul `current` indică PID-ul procesului curent. Următoarele funcții se consideră implementate la nivelul nucleului și pot fi folosite pentru rezolvarea problemei:

- `void lock(lock_t the_lock);`
- `void unlock(lock_t the_lock);`
- `void initlock(lock_t the_lock);`
- `void enable_intr();`
- `void disable_intr();`
- `int copy_to_user(void *from, void *to, int length);`
- `int copy_from_user(void *from, void *to, int length);`
- `void schedule(void);`
- `void *kmalloc(int size);`

- `void printk(const char *, ...);`

Notă: nu este necesar să utilizați *toate* aceste funcții.

2. [0.75 puncte] Un sistem de operare folosește următoarele stări pentru procese:

JUST_CREATED, READY, RUNNING, BLOCKED, ZOMBIE.

Trei procese din acest sistem trec, pe parcursul existenței lor, prin următoarele stări (succesiv de la stânga la dreapta):

- (a) JUST_CREATED, RUNNING, BLOCKED, READY, RUNNING, ZOMBIE;
- (b) JUST_CREATED, BLOCKED, READY, RUNNING, ZOMBIE;
- (c) READY, RUNNING, READY, RUNNING, READY, BLOCKED, READY, RUNNING, ZOMBIE;

Secvențele de mai sus conțin un număr de greșeli. Identificați-le și precizați, pe scurt, motivul erorii.

3. [0.75 puncte] Un proces are primele intrări în tabela de pagini după cum urmează:

Virt. page	Phys. address	Valid
1	12288(0x3000)	yes
2	00000(0x0000)	yes
3	20480(0x5000)	yes
4	30222(0x760E)	no
5	08192(0x2000)	yes

Știind că sistemul folosește doar paginare, cu dimensiunea paginii de 4kB și cea a adresei de 16 biți, să se precizeze:

- Numărul de pagini virtuale pe care le poate adresa un proces
- În ce pagină se găsește octetul cu adresa 8349 (0x209D)
- La ce adresa fizică se găsește octetul de mai sus