

③7. MONITORY

- jedna se o abstraktní datový typ, či datovou strukturu
- reprezentuje se sobě sdílená data se podobě sdílených proměnných, ke kterým třeba muset být rovněž výlučný přístup a tedy vzájemné vyloučení
- nad tímto daty jsou různé atomické operace, které tato data čtou a manipuluje s nimi a upravují je atd.
- z hlediska toho, že s datům monitor musel být vzájemné vyloučení, se pracuje tak před vstupem do monitoru a vyloučením některé operace nad daty musí proces získat právo
- pokud je v monitoru již proces tak tento se chce vstoupit je pozastaven při pokusu o získání práva a je na frontě čekajících procesů na právo a je připnut kódek
- ty atomické operace jsou z hlediska spirosternu jiné přístupové k datům monitoru  
⇒ nová rozbíjení monitoru
- reálné právo při vstupu do operace je realizováno automaticky a programista to nemusí řídit, to samé s vyloučením některé operace → to udělat se práce s monitorem je muset počítat je ale ještě méně abstrakce
- ty samé jsou realizovány pomocí binárních semaforů, konkrétní metody
- novitě monitor s nějakou operací je možné mít potřebu čekat na splnění nějakých podmínek
- třeba proces musí splňovat nějaké podmínky předtím, než se může vstoupit do monitoru
- s tím se používají podmíněné proměnné v monitoru
- pokud je podmínka splněna, tak proces pokračuje dál, na té podmínce, condition, fronta čekajících na podmínku
- pokud podmínka není splněna, tak proces rovněž c.wait() a je vložen do fronty procesů čekajících na té podmínce a možná právo monitoru a je mu připnut kódek
- do monitoru tak mohou vstupovat další procesy, protože se možná právo
- jakmile nějaký proces splní podmínku tak může rovněž c.signal() a tím odvoláji frontu čekajících procesů na té c frontě → ještě problém: ( - tedy jsou v monitoru dva procesy : ten co volá a ten co se probouzí a to se nemusí stát

2. рѣшеніе

House + Hansen - to modrin' (c. signu(1)) ten modrin'ci'  
- bloliz'ci' signatiane proces poverun' a dute probel' ten modrin' ci'  
ci' obelene fuzi a mariton, nelo pame nide  
celat a modin' celat ten modrin'ci' move' proces  
co celat' ten roshup do mariton ale ten modrin'ci'  
ampron + Redell - ten co modin' lize d'el a ten modrin'ci' amon  
celat' na pindelen' celat' jize a nion ten celat'  
i modrin'ci' proces co celat'ci' lepe roshupit do  
mariton a co modrin'ci' shon' a mariton (celat'ci') nelo  
pame nide celat' kel but poverun' ten modrin'ci' nelo jiz' a



- keď keď bude dáta podozriť, že sa jej podarí na podmínku a bude nájdená, napíše jej, že je čas do monitora prejsť a odpovedať
- keď monitor, musí byť rovnako otestovať podmínku a keď sa na mi čísel podmienka zmení, píše si, že tam je čas na monitor a napíše jej, že je čas do monitora prejsť a podmínku sa zmeniť
- keď prvýkrát odpovie signal() ale notify()

- monitor keď musí:

- 1) odčítať dáta / odčítať podmienku
- 2) nájsť / nájsť - to POSIX pthread-mutex-t
- 3) podmínku (preto čítajúci proces na podmínku) - to POSIX pthread-cond-t

- používa sa signal() / notify() a wait()

pthread-cond-t ci

c.wait(); c.signal(); c.notify();

pthread-cond-wait(c); pthread-cond-signal(c)

- to UNIXový systémový rozhranie POSIX robí (pthread-t) najmä oboje

semafory ale môžu byť aj používajú monitor

- keď dáta sú v monitoru, budúce dáta sa odčítajú, podmienka sa zmení a odčítajú sa operácie, ktoré a robia.

Pt:

int suma = 0;

pthread\_mutex\_lock(&mutex);

while (suma <= 0) {

pthread\_cond\_wait(&cond);

}

-- spracovanie

pthread\_mutex\_unlock(&mutex);

ČEKÁNÍ V MONITORE NA SPRÁVENÍ PODMIENKY

podmienka podmienka

musí sa do monitora a volá sa do podmienky podmienky, ktoré je to prvýkrát, keď sa zmení podmienka a keď podmienka je to prvýkrát, keď sa zmení podmienka a keď podmienka je to prvýkrát, keď sa zmení podmienka

musí sa zmeniť (mutex) a robí sa to prvýkrát, keď sa zmení podmienka a keď podmienka je to prvýkrát, keď sa zmení podmienka

SIGNALIZÁCIA SPRÁVENÍ PODMIENKY

pthread\_mutex\_lock(&mutex);

suma += 100; -- prvý spracovanie

pthread\_cond\_signal(&cond);

pthread\_mutex\_unlock(&mutex);

musí sa zmeniť podmienku a keď podmienka je to prvýkrát, keď sa zmení podmienka

musí sa zmeniť podmienku a keď podmienka je to prvýkrát, keď sa zmení podmienka

- monitor sa používa, keď je potrebné získať, keď je potrebné získať, keď je potrebné získať

- na dáta v monitoroch je to prvýkrát, keď je potrebné získať

- programátor je to prvýkrát, keď je potrebné získať

- to monitor môže odčítať dáta - čítajú na podmínku - robí sa monitor

monitor < operácie sa odčítajú podmienkami

Čekanie na podmínku - wait() a signalizácia signal() / notify()

(2)