

PPADS MMXX

Matus Jokay, C-503, matus.jokay@stuba.sk

Roderik Ploszek, C-512, roderik.ploszek@stuba.sk

uim.fei.stuba.sk/predmet/i-ppds

konzultacie dohodou

✘ Vyhľadovanie

✘ Priorita kategórie procesov

VYHLADOVENIE

Init():

readLS $\leftarrow$ LS()

roomE $\leftarrow$ Sem(1)

R():

1) readLS.lock(roomE)

2) // stale je tu nejaky cit.

3) readLS.unlock(roomE)

W():

1) roomE.wait()

2) // kod zapis.

3) roomE.signal()

VYHLADOVENIE

Init():

`readLS` $\leftarrow$ `LS()`

`roomE` $\leftarrow$ `Sem(1)`

`R():`

- 1) `readLS.lock(roomE)`
- 2) `// stale je tu nejaky cit.`
- 3) `readLS.unlock(roomE)`

- Pridame turniket, cez ktory musia R prechadzat
- Uzamykat ho budu W

VYHLADOVENIE

Init():
readLS $\leftarrow$ LS()
roomE $\leftarrow$ Sem(1)
turn $\leftarrow$ Sem(1)

R():

- 1)
- 2)
- 3) readLS.lock(roomE)
- 4) // kod citatelov
- 5) readLS.unlock(roomE)

W():

- 1)
- 2) roomE.wait()
- 3) // kod zapis.
- 4) roomE.signal()
- 5)

VYHLADOVENIE

Init():
readLS $\leftarrow$ LS()
roomE $\leftarrow$ Sem(1)
turn $\leftarrow$ Sem(1)

R():

- 1) turn.wait()
- 2) turn.signal()
- 3) readLS.lock(roomE)
- 4) // kod citatelov
- 5) readLS.unlock(roomE)

W():

- 1) turn.wait()
- 2) roomE.wait()
- 3) // kod zapis.
- 4) roomE.signal()
- 5) turn.signal()

VYHLADOVENIE

Init():
readLS $\leftarrow$ LS()
roomE $\leftarrow$ Sem(1)
turn $\leftarrow$ Sem(1)

W():

- 1) turn.wait()
- 2) roomE.wait()
- 3) // kod zapis.
- 4) turn.signal()
- 5) roomE.signal()

W():

- 1) turn.wait()
- 2) roomE.wait()
- 3) // kod zapis.
- 4) roomE.signal()
- 5) turn.signal()

AKY JE ROZDIEL MEDZI TYMITO VERZIAMI?

W():

- 1) `turn.wait()`
- 2) `roomE.wait()`
- 3) `// kod zapis.`
- 4) `turn.signal()`
- 5) `roomE.signal()`



W():

- 1) `turn.wait()`
- 2) `roomE.wait()`
- 3) `// kod zapis.`
- 4) `roomE.signal()`
- 5) `turn.signal()`

Init():
readLS $\leftarrow$ LS()
roomE $\leftarrow$ Sem(1)
turn $\leftarrow$ Sem(1)

R():

- 1) turn.wait()
- 2) turn.signal()
- 3) readLS.lock(roomE)
- 4) // kod citatelov
- 5) readLS.unlock(roomE)

W_ver1():

- 1) turn.wait()
- 2) roomE.wait()
- 3) // kod zapis.
- 4) turn.signal()
- 5) roomE.signal()

W_ver2():

- 1) turn.wait()
- 2) roomE.wait()
- 3) // kod zapis.
- 4) roomE.signal()
- 5) turn.signal()

<pre> Init(): readLS ← LS() roomE ← Sem(1) turn ← Sem(1) </pre>	<pre> W_ver1(): 1) turn.wait() 2) roomE.wait() 3) // kod zapis. 4) turn.signal() 5) roomE.signal() </pre>	<pre> W_ver2(): 1) turn.wait() 2) roomE.wait() 3) // kod zapis. 4) roomE.signal() 5) turn.signal() </pre>
<pre> R(): 1) turn.wait() 2) turn.signal() 3) readLS.lock(roomE) 4) // kod citatelov 5) readLS.unlock(roomE) </pre>		

- ✘ Pri rieseni R/W predpokladame silne semafore, t.j. take, ktore implementuju FIFO frontu na cakanie
- ✘ Pri oboch verziach W() sa citatelia ukladaju do FIFO fronty na turnikete
- ✘ Citatel moze pokracovat az po uvolneni turniketu

<u>Init():</u> readLS $\leftarrow$ LS() roomE $\leftarrow$ Sem(1) turn $\leftarrow$ Sem(1)	W_ver1(): 1) turn.wait() 2) roomE.wait() 3) // kod zapis. 4) turn.signal() 5) roomE.signal()	W_ver2(): 1) turn.wait() 2) roomE.wait() 3) // kod zapis. 4) roomE.signal() 5) turn.signal()
R(): 1) turn.wait() 2) turn.signal() 3) readLS.lock(roomE) 4) // kod citatelov 5) readLS.unlock(roomE)		

- ✗ Vo verzii 1 sa turniket odomyka este pred “odidenim” zapisovatela z miestnosti, takže
 - + Po vykonani W_ver1() riadku 4 moze pokracovat citatel cez turniket
 - + Akonahle prejde cez turniket, pokusi sa ziskat exkluzivny pristup do miestnosti (kedze sa jedna o prveho citatela prejdúvsieho cez turniket)
 - + To sa mu vsak nepodari, musi cakat a znovu moze ist jedine W_ver1(), ktore az na riadku 5 uvolnuje miestnost

<u>Init():</u> readLS ← LS() roomE ← Sem(1) turn ← Sem(1)	W_ver1(): 1) turn.wait() 2) roomE.wait() 3) // kod zapis. 4) turn.signal() 5) roomE.signal()	W_ver2(): 1) turn.wait() 2) roomE.wait() 3) // kod zapis. 4) roomE.signal() 5) turn.signal()
R(): 1) turn.wait() 2) turn.signal() 3) readLS.lock(roomE) 4) // kod citatelov 5) readLS.unlock(roomE)		

- ✘ Vo verzii 2 sa turniket odomyka az po opusteni miestnosti, takze:
 - + Akonahle W_ver2() vykona riadok 5, miestnost je prazdna
 - + Citatel moze prejsť turniketom a získať exkluzívny prístup do miestnosti pre citateľov
 - + Nie je potrebné ďalšie preplanovanie vlákna/procesu navyše

<u>Init():</u> readLS $\leftarrow$ LS() roomE $\leftarrow$ Sem(1) turn $\leftarrow$ Sem(1)	W_ver1(): 1) turn.wait() 2) roomE.wait() 3) // kod zapis. 4) turn.signal() 5) roomE.signal()	W_ver2(): 1) turn.wait() 2) roomE.wait() 3) // kod zapis. 4) roomE.signal() 5) turn.signal()
R(): 1) turn.wait() 2) turn.signal() 3) readLS.lock(roomE) 4) // kod citatelov 5) readLS.unlock(roomE)		

- ✘ Vo verzii 2 sa turniket odomyka az po opusteni miestnosti, takze:
 - + Akonahle W_ver2() vykona riadok 5, miestnost je prazdna
 - + Citatel moze prejsť turniketom a získať exkluzívny prístup do miestnosti pre citateľov
 - + Nie je potrebné ďalšie preplanovanie vlákna/procesu navyše

- ✘ Preto je verzia 2 efektívnejšia

PRIORITA KATEGORIE PROCESOV

- ✗ Vyriesili sme vyhladovenie
- ✗ Co ked potrebujeme prioritu jednej kategorie (zapisovatelov)?
- ✗ Priklad: zapisovatelia, ked aktualizuju hodnotu cidiel v atomovej elektrarni
 - + Cim menej citatelov by malo citat, ked zapisovatel chce aktualizovat udaje, aby rychlo opustili oblast citania, a zapisovatel mohol aktualizovat udaje...

PRIORITA KATEGORIE PROCESOV

- ✗ V nasom rieseni R-W mame semafor pomenovany **turn**
- ✗ Premenujme ho na **noReaders**
 - + Jeho vyznam je, aby zapisovatelia dokazali zabranit citatelom vstup do miestnosti
 - + Preto jeho meno by malo odzrkadlovat tento vyznam

PRIORITA ZAPISOVATELOV

Init():
readLS $\leftarrow$ LS()
roomE $\leftarrow$ Sem(1)
turn $\leftarrow$ Sem(1)

R():

- 1) turn.wait()
- 2) turn.signal()
- 3) readLS.lock(roomE)
- 4) // kod citatelov
- 5) readLS.unlock(roomE)

W():

- 1) turn.wait()
- 2) roomE.wait()
- 3) // kod zapis.
- 4) roomE.signal()
- 5) turn.signal()

PRIORITA ZAPISOVATELOV

Init():

readLS $\leftarrow$ LS()

roomE $\leftarrow$ Sem(1)

noReaders $\leftarrow$ Sem(1)

R():

- 1) noReaders.wait()
- 2) noReaders.signal()
- 3) readLS.lock(roomE)
- 4) // kod citatelov
- 5) readLS.unlock(roomE)

W():

- 1) noReaders.wait()
- 2) roomE.wait()
- 3) // kod zapis.
- 4) roomE.signal()
- 5) noReaders.signal()

PRIORITA KATEGORIE PROCESOV

- ✗ Podobne premenujeme semafor **roomE** na **noWriters**
- ✗ Jeho význam
 - + Pre čitateľov znamená, že jeho držanie zabráňuje zapisovateľom vstup do miestnosti
 - + Pre zapisovateľa znamená, že zabráňuje vstup do miestnosti iným zapisovateľom
- ✗ Zabráňuje vstup zapisovateľom

PRIORITA ZAPISOVATELOV

Init():

readLS $\leftarrow$ LS()

roomE $\leftarrow$ Sem(1)

noReaders $\leftarrow$ Sem(1)

R():

- 1) noReaders.wait()
- 2) noReaders.signal()
- 3) readLS.lock(roomE)
- 4) // kod citatelov
- 5) readLS.unlock(roomE)

W():

- 1) noReaders.wait()
- 2) roomE.wait()
- 3) // kod zapis.
- 4) roomE.signal()
- 5) noReaders.signal()

PRIORITA ZAPISOVATELOV

```
Init():  
    readLS ← LS()  
    noWriters ← Sem(1)  
    noReaders ← Sem(1)
```

R():

- 1) noReaders.wait()
- 2) noReaders.signal()
- 3) readLS.lock(noWriters)
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

W():

- 1) noReaders.wait()
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) noReaders.signal()

PRIORITA KATEGORIE PROCESOV

- ✗ Zatial mame to iste riesenie ako pri vyhľadovaní
- ✗ Premenovali sme premenne
- ✗ Ako zabezpečiť prioritu zapisovateľom?

PRIORITA ZAPISOVATELOV

```
Init():  
    readLS ← LS()  
    noWriters ← Sem(1)  
    noReaders ← Sem(1)
```

R():

- 1) noReaders.wait()
- 2) noReaders.signal()
- 3) readLS.lock(noWriters)
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

W():

- 1) noReaders.wait()
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) noReaders.signal()

PRIORITA ZAPISOVATELOV

Init():

readLS $\leftarrow$ LS()

noWriters $\leftarrow$ Sem(1)

noReaders $\leftarrow$ Sem(1)

R():

- 1) noReaders.wait()
- 2) noReaders.signal()
- 3) readLS.lock(noWriters)
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

W():

- 1) noReaders.wait()
- 2) noWriters.wait()
- 3) // kod zapisu
- 4) noWriters.signal()
- 5) noReaders.signal()

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `noReaders.signal()`
- 3) `readLS.lock(noWriters)`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `noReaders.signal()`
- 3) `readLS.lock(noWriters)`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

R():

- 1) `noReaders.wait()`
- 2) `noReaders.signal()`
- 3) `readLS.lock(noWriters)`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

PRIORITA ZAPISOVATELOV

R():

- 1) noReaders.wait()
- 2) noReaders.signal()
- 3) readLS.lock(noWriters)
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

R():

- 1) noReaders.wait()
- 2) noReaders.signal()
- 3) readLS.lock(noWriters)
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

PRIORITA ZAPISOVATELOV

R():

- 1) noReaders.wait()
- 2) noReaders.signal()
- 3) readLS.lock(noWriters)
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

R():

- 1) noReaders.wait()
- 2) noReaders.signal()
- 3) readLS.lock(noWriters)
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

PRIORITA ZAPISOVATELOV

R():

- 1) noReaders.wait()
- 2) noReaders.signal()
- 3) readLS.lock(noWriters)
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

R():

- 1) noReaders.wait()
- 2) readLS.lock(noWriters)
- 3) noReaders.signal()
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

PRIORITA ZAPISOVATELOV

R():

- 1) noReaders.wait()
- 2) noReaders.signal()
- 3) readLS.lock(noWriters)
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

R():

- 1) noReaders.wait()
- 2) readLS.lock(noWriters)
- 3) noReaders.signal()
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

PRIORITA ZAPISOVATELOV

R():

- 1) noReaders.wait()
- 2) noReaders.signal()
- 3) readLS.lock(noWriters)
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

R():

- 1) noReaders.wait()
- 2) readLS.lock(noWriters)
- 3) noReaders.signal()
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

PRIORITA ZAPISOVATELOV

R():

- 1) noReaders.wait()
- 2) noReaders.signal()
- 3) readLS.lock(noWriters)
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

R():

- 1) noReaders.wait()
- 2) readLS.lock(noWriters)
- 3) noReaders.signal()
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

PRIORITA ZAPISOVATELOV

R():

- 1) noReaders.wait()
- 2) noReaders.signal()
- 3) readLS.lock(noWriters)
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

W():

- 1) noReaders.wait()
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) noReaders.signal()

R():

- 1) noReaders.wait()
- 2) readLS.lock(noWriters)
- 3) noReaders.signal()
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

W():

- 1) noReaders.wait()
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) noReaders.signal()

PRIORITA ZAPISOVATELOV

R():

- 1) noReaders.wait()
- 2) noReaders.signal()
- 3) readLS.lock(noWriters)
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

W():

- 1) noReaders.wait()
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) noReaders.signal()

R():

- 1) noReaders.wait()
- 2) readLS.lock(noWriters)
- 3) noReaders.signal()
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

W():

- 1) noReaders.wait()
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) noReaders.signal()

PRIORITA ZAPISOVATELOV

R():

- 1) noReaders.wait()
- 2) noReaders.signal()
- 3) readLS.lock(noWriters)
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

W():

- 1) noReaders.wait()
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) noReaders.signal()

R():

- 1) noReaders.wait()
- 2) readLS.lock(noWriters)
- 3) noReaders.signal()
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

W():

- 1) noReaders.wait()
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) noReaders.signal()

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `noReaders.signal()`
- 3) `readLS.lock(noWriters)`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

W():

- 1) `noReaders.wait()`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `noReaders.signal()`

PRIORITA ZAPISOVATELOV

R():

riesenie vlavo nezarucuje, aby r() nebol
predbehnutý w()

- 1) `noReaders.wait()`
- 2) `noReaders.signal()`
- 3) `readLS.lock(noWriters)`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

W():

- 1) `noReaders.wait()`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `noReaders.signal()`

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `noReaders.signal()`
- 3) `readLS.lock(noWriters)`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

1) nech w1 drzi noWriters, zapisuje; zaroven drzi noReaders

W():

- 1) `noReaders.wait()`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `noReaders.signal()`

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `noReaders.signal()`
- 3) `readLS.lock(noWriters)`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

1) nech w1 drzi noWriters, zapisuje; zaroven drzi noReaders

2) nech sa pocas zapisu w1 na noReaders radia r1, w2, ...

W():

- 1) `noReaders.wait()`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `noReaders.signal()`

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `noReaders.signal()`
- 3) `readLS.lock(noWriters)`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

1) nech w1 drzi noWriters, zapisuje; zaroven drzi noReaders

2) nech sa pocas zapisu w1 na noReaders radia r1, w2, ...

3) nech w1 uvolni noWriters (pokracovat nemoze nikto, este stale w1 drzi noReaders, a tam su vsetci blokovani)

W():

- 1) `noReaders.wait()`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `noReaders.signal()`

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `noReaders.signal()`
- 3) `readLS.lock(noWriters)`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

W():

- 1) `noReaders.wait()`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `noReaders.signal()`

1) nech w1 drzi noWriters, zapisuje; zaroven drzi noReaders

2) nech sa pocas zapisu w1 na noReaders radia r1, w2, ...

3) nech w1 uvolni noWriters (pokracovat nemoze nikto, este stale w1 drzi noReaders, a tam su vsetci blokovani)

4) nech w1 uvolni noReaders; pokracovat moze iba r1, lebo ostatni su za nim vo FIFO

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `noReaders.signal()`
- 3) `readLS.lock(noWriters)`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

W():

- 1) `noReaders.wait()`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `noReaders.signal()`

- 1) nech w1 drzi noWriters, zapisuje; zaroven drzi noReaders
- 2) nech sa pocas zapisu w1 na noReaders radia r1, w2, ...
- 3) nech w1 uvolni noWriters (pokracovat nemoze nikto, este stale w1 drzi noReaders, a tam su vsetci blokovani)
- 4) nech w1 uvolni noReaders; pokracovat moze iba r1, lebo ostatni su za nim vo FIFO
- 5) r1 zoberie noReaders, ale hned ho aj uvolni

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `noReaders.signal()`
- 3) `readLS.lock(noWriters)`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

W():

- 1) `noReaders.wait()`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `noReaders.signal()`

- 1) nech w1 drzi noWriters, zapisuje; zaroven drzi noReaders
- 2) nech sa pocas zapisu w1 na noReaders radia r1, w2, ...
- 3) nech w1 uvolni noWriters (pokracovat nemoze nikto, este stale w1 drzi noReaders, a tam su vsetci blokovani)
- 4) nech w1 uvolni noReaders; pokracovat moze iba r1, lebo ostatni su za nim vo FIFO
- 5) r1 zoberie noReaders, ale hned ho aj uvolni
- 6) v tejto chvíli moze pokracovat ako r1 tak aj w2; w2 moze zobrat noWriters, a tym predbehne r1

PRIORITA ZAPISOVATELOV

R():

- 1) noReaders.wait()
- 2) noReaders.signal()
- 3) readLS.lock(noWriters)
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

W():

- 1) noReaders.wait()
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) noReaders.signal()

R():

- 1) noReaders.wait()
- 2) readLS.lock(noWriters)
- 3) noReaders.signal()
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

W():

- 1) noReaders.wait()
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) noReaders.signal()

PRIORITA ZAPISOVATELOV

R():

- 1) noReaders.wait()
- 2) noReaders.signal()
- 3) readLS.lock(noWriters)
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

W():

- 1) noReaders.wait()
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) noReaders.signal()

R():

- 1) noReaders.wait()
- 2) readLS.lock(noWriters)
- 3) noReaders.signal()
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

W():

- 1) noReaders.wait()
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) noReaders.signal()

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `readLS.lock(noWriters)`
- 3) `noReaders.signal()`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

W():

- 1) `noReaders.wait()`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `noReaders.signal()`

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `readLS.lock(noWriters)`
- 3) `noReaders.signal()`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

W():

- 1) `noReaders.wait()`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `noReaders.signal()`

✗ Semafor `noReaders` ma FIFO frontu

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `readLS.lock(noWriters)`
- 3) `noReaders.signal()`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

W():

- 1) `noReaders.wait()`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `noReaders.signal()`

- ✗ Semafor `noReaders` ma FIFO frontu
- + Ak `w()` prejde, `r()` uz sa moze zaradit

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `readLS.lock(noWriters)`
- 3) `noReaders.signal()`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

W():

- 1) `noReaders.wait()`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `noReaders.signal()`

- ✗ Semafor `noReaders` ma FIFO frontu
 - + Ak `w()` prejde, `r()` uz sa moze zaradit
 - + Dalsi `w()` moze ist do fronty az za `r()`

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `readLS.lock(noWriters)`
- 3) `noReaders.signal()`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

W():

- 1) `noReaders.wait()`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `noReaders.signal()`

- ✗ Semafor `noReaders` ma FIFO frontu
 - + Ak `w()` prejde, `r()` uz sa moze zaradit
 - + Dalsi `w()` moze ist do fronty az za `r()`
 - + Takze po skonceni `w()` ide `r()`, az potom dalsi `w()`

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `readLS.lock(noWriters)`
- 3) `noReaders.signal()`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

W():

- 1) `noReaders.wait()`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `noReaders.signal()`

✗ Ako to opravit?

✗ Co znamena oprava?

+ Vediet menit radenie vo fronte na semafor...

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `readLS.lock(noWriters)`
- 3) `noReaders.signal()`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

W():

- 1) `noReaders.wait()`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `noReaders.signal()`

✗ Modifikovat OS?

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `readLS.lock(noWriters)`
- 3) `noReaders.signal()`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

W():

- 1) `noReaders.wait()`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `noReaders.signal()`

✗ Modifikovat OS? Pfuuuuuuuuuuuuuuj!!!

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `readLS.lock(noWriters)`
- 3) `noReaders.signal()`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

W():

- 1) `noReaders.wait()`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `noReaders.signal()`

✗ Co robi ADT LS?

PRIORITA ZAPISOVATELOV

R():

- 1) noReaders.wait()
- 2) readLS.lock(noWriters)
- 3) noReaders.signal()
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

W():

- 1) noReaders.wait()
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) noReaders.signal()

PRIORITA ZAPISOVATELOV

W():

- 1) noReaders.wait()
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) noReaders.signal()

W():

- 1) noReaders.wait()
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) noReaders.signal()

PRIORITA ZAPISOVATELOV

W():

- 1) noReaders.wait()
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) noReaders.signal()

W():

- 1) noReaders.wait()
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) noReaders.signal()

PRIORITA ZAPISOVATELOV

W():

- 1) noReaders.wait()
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) noReaders.signal()

W():

- 1) writeLS.lock(noReaders)
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) writeLS.unlock(noReaders)

PRIORITA ZAPISOVATELOV

W():

- 1) `writeLS.lock(noReaders)`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `writeLS.unlock(noReaders)`

PRIORITA ZAPISOVATELOV

W():

- 1) `writeLS.lock(noReaders)`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `writeLS.unlock(noReaders)`

✘ Ake bude teda cele riesenie?

PRIORITA ZAPISOVATELOV

R():

- 1) noReaders.wait()
- 2) readLS.lock(noWriters)
- 3) noReaders.signal()
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

W():

- 1) writeLS.lock(noReaders)
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) writeLS.unlock(noReaders)

Init():

```
readLS ← LS()
writeLS ← LS()
noWriters ← Sem(1)
noReaders ← Sem(1)
```

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `readLS.lock(noWriters)`
- 3) `noReaders.signal()`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

W():

- 1) `writeLS.lock(noReaders)`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `writeLS.unlock(noReaders)`

✖ Poznamka

+ No starvation pre w() sme poriesili, ale...

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `readLS.lock(noWriters)`
- 3) `noReaders.signal()`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

W():

- 1) `writeLS.lock(noReaders)`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `writeLS.unlock(noReaders)`

✖ Poznámka

- + No starvation pre w() sme poriesili, ale...
- + Zaviedli sme moznost starvation pre r() ;)

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `readLS.lock(noWriters)`
- 3) `noReaders.signal()`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

W():

- 1) `writeLS.lock(noReaders)`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `writeLS.unlock(noReaders)`

✖ Poznamka

+ Ako poriesit no-starvation pre r()?

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `readLS.lock(noWriters)`
- 3) `noReaders.signal()`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

W():

- 1) `writeLS.lock(noReaders)`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `writeLS.unlock(noReaders)`

✖ Poznámka

- + Ako poriesit no-starvation pre r()?
- + Prioritou pre r()!

PRIORITA ZAPISOVATELOV

R():

- 1) `noReaders.wait()`
- 2) `readLS.lock(noWriters)`
- 3) `noReaders.signal()`
- 4) `// kod citatelov`
- 5) `readLS.unlock(noWriters)`

W():

- 1) `writeLS.lock(noReaders)`
- 2) `noWriters.wait()`
- 3) `// kod zapis.`
- 4) `noWriters.signal()`
- 5) `writeLS.unlock(noReaders)`

✗ Poznamka

- + Ako poriesit no-starvation pre r()?
- + Pridame teda turniket?

PRIORITA CITATELOV

R():

- 1) noReaders.wait()
- 2) readLS.lock(noWriters)
- 3) noReaders.signal()
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

W():

- 1) writeLS.lock(noReaders)
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) writeLS.unlock(noReaders)

PRIORITA CITATELOV

R():

- 1) noReaders.wait()
- 2) readLS.lock(noWriters)
- 3) noReaders.signal()
- 4) // kod citatelov
- 5) readLS.unlock(noWriters)

W():

- 1) writeLS.lock(noReaders)
- 2) noWriters.wait()
- 3) // kod zapis.
- 4) noWriters.signal()
- 5) writeLS.unlock(noReaders)

PRIORITA CITATELOV

R():

- 1)
- 2) `noReaders.wait()`
- 3) `readLS.lock(noWriters)`
- 4) `noReaders.signal()`
- 5) `// kod citatelov`
- 6) `readLS.unlock(noWriters)`
- 7)

W():

- 1)
- 2) `writeLS.lock(noReaders)`
- 3)
- 4) `noWriters.wait()`
- 5) `// kod zapis.`
- 6) `noWriters.signal()`
- 7) `writeLS.unlock(noReaders)`

PRIORITA CITATELOV

R():

- 1) `_LS().lock(noStarvR)`
- 2) `noReaders.wait()`
- 3) `readLS.lock(noWriters)`
- 4) `noReaders.signal()`
- 5) `// kod citatelov`
- 6) `readLS.unlock(noWriters)`
- 7) `_LS().unlock(noStarvR)`

W():

- 1) `noStarvR.wait()`
- 2) `writeLS.lock(noReaders)`
- 3) `noStarvR.signal()`
- 4) `noWriters.wait()`
- 5) `// kod zapis.`
- 6) `noWriters.signal()`
- 7) `writeLS.unlock(noReaders)`

PRIORITA CITATELOV

R():

- 1) `_LS().lock(noStarvR)`
- 2) `noReaders.wait()`
- 3) `readLS.lock(noWriters)`
- 4) `noReaders.signal()`
- 5) `// kod citatelov`
- 6) `readLS.unlock(noWriters)`
- 7) `_LS().unlock(noStarvR)`

W():

- 1) `noStarvR.wait()`
- 2) `writeLS.lock(noReaders)`
- 3) `noStarvR.signal()`
- 4) `noWriters.wait()`
- 5) `// kod zapis.`
- 6) `noWriters.signal()`
- 7) `writeLS.unlock(noReaders)`

✗ Co sme to len urobili?

PRIORITA CITATELOV

R():

- 1) `_LS().lock(noStarvR)`
- 2) `noReaders.wait()`
- 3) `readLS.lock(noWriters)`
- 4) `noReaders.signal()`
- 5) `// kod citatelov`
- 6) `readLS.unlock(noWriters)`
- 7) `_LS().unlock(noStarvR)`

W():

- 1) `noStarvR.wait()`
- 2) `writeLS.lock(noReaders)`
- 3) `noStarvR.signal()`
- 4) `noWriters.wait()`
- 5) `// kod zapis.`
- 6) `noWriters.signal()`
- 7) `writeLS.unlock(noReaders)`

✘ Semafor noReaders akosi postrada svoj význam...

PRIORITA CITATELOV

R():

- 1) `_LS().lock(noStarvR)`
- 2) `noReaders.wait()`
- 3) `readLS.lock(noWriters)`
- 4) `noReaders.signal()`
- 5) `// kod citatelov`
- 6) `readLS.unlock(noWriters)`
- 7) `_LS().unlock(noStarvR)`

W():

- 1) `noStarvR.wait()`
- 2) `writeLS.lock(noReaders)`
- 3) `noStarvR.signal()`
- 4) `noWriters.wait()`
- 5) `// kod zapis.`
- 6) `noWriters.signal()`
- 7) `writeLS.unlock(noReaders)`

✗ Takze sa ho mozeme zbavit

PRIORITA CITATELOV

R():

- 1) `_LS().lock(noStarvR)`
- 2)
- 3) `readLS.lock(noWriters)`
- 4)
- 5) `// kod citatelov`
- 6) `readLS.unlock(noWriters)`
- 7) `_LS().unlock(noStarvR)`

W():

- 1) `noStarvR.wait()`
- 2)
- 3) `noStarvR.signal()`
- 4) `noWriters.wait()`
- 5) `// kod zapis.`
- 6) `noWriters.signal()`
- 7)

✗ Takze sa ho mozeme zbavit

PRIORITA CITATELOV

R():

- 1) `_LS().lock(noStarvR)`
- 2)
- 3) `readLS.lock(noWriters)`
- 4)
- 5) `// kod citatelov`
- 6) `readLS.unlock(noWriters)`
- 7) `_LS().unlock(noStarvR)`

W():

- 1) `noStarvR.wait()`
- 2)
- 3) `noStarvR.signal()`
- 4) `noWriters.wait()`
- 5) `// kod zapis.`
- 6) `noWriters.signal()`
- 7)

✗ Upravime si kodik

PRIORITA CITATELOV

R():

- 1) `_LS().lock(noStarvR)`
- 2) `readLS.lock(noWriters)`
- 3) `// kod citatelov`
- 4) `readLS.unlock(noWriters)`
- 5) `_LS().unlock(noStarvR)`
- 6)
- 7)

W():

- 1) `noStarvR.wait()`
- 2) `noStarvR.signal()`
- 3) `noWriters.wait()`
- 4) `// kod zapis.`
- 5) `noWriters.signal()`
- 6)
- 7)

PRIORITA CITATELOV

R():

- 1) `_LS().lock(noStarvR)`
- 2) `readLS.lock(noWriters)`
- 3) `// kod citatelov`
- 4) `readLS.unlock(noWriters)`
- 5) `_LS().unlock(noStarvR)`
- 6)
- 7)

W():

- 1) `noStarvR.wait()`
- 2) `noStarvR.signal()`
- 3) `noWriters.wait()`
- 4) `// kod zapis.`
- 5) `noWriters.signal()`
- 6)
- 7)

✗ A co vidime?

PRIORITA CITATELOV

R():

- 1) `_LS().lock(noStarvR)`
- 2) `readLS.lock(noWriters)`
- 3) `// kod citatelov`
- 4) `readLS.unlock(noWriters)`
- 5) `_LS().unlock(noStarvR)`
- 6)
- 7)

W():

- 1) `noStarvR.wait()`
- 2) `noStarvR.signal()`
- 3) `noWriters.wait()`
- 4) `// kod zapis.`
- 5) `noWriters.signal()`
- 6)
- 7)

✗ 2x LS po sebe...

PRIORITA CITATELOV

R():

- 1) `_LS().lock(noStarvR)`
- 2) `readLS.lock(noWriters)`
- 3) `// kod citatelov`
- 4) `readLS.unlock(noWriters)`
- 5) `_LS().unlock(noStarvR)`
- 6)
- 7)

W():

- 1) `noStarvR.wait()`
- 2) `noStarvR.signal()`
- 3) `noWriters.wait()`
- 4) `// kod zapis.`
- 5) `noWriters.signal()`
- 6)
- 7)

✗ 2x LS po sebe... dost zbytocne

PRIORITA CITATELOV

R():

- 1) `_LS().lock(noStarvR)`
- 2) `readLS.lock(noWriters)`
- 3) `// kod citatelov`
- 4) `readLS.unlock(noWriters)`
- 5) `_LS().unlock(noStarvR)`
- 6)
- 7)

W():

- 1) `noStarvR.wait()`
- 2) `noStarvR.signal()`
- 3) `noWriters.wait()`
- 4) `// kod zapis.`
- 5) `noWriters.signal()`
- 6)
- 7)

✗ Takze znovu zjednodusime

PRIORITA CITATELOV

R():

- 1)
- 2) `readLS.lock(noWriters)`
- 3) `// kod citatelov`
- 4) `readLS.unlock(noWriters)`
- 5)
- 6)
- 7)

W():

- 1)
- 2)
- 3) `noWriters.wait()`
- 4) `// kod zapis.`
- 5) `noWriters.signal()`
- 6)
- 7)

✗ Takze znovu zjednodusime

PRIORITA CITATELOV

R():

- 1)
- 2) `readLS.lock(noWriters)`
- 3) `// kod citatelov`
- 4) `readLS.unlock(noWriters)`
- 5)
- 6)
- 7)

W():

- 1)
- 2)
- 3) `noWriters.wait()`
- 4) `// kod zapis.`
- 5) `noWriters.signal()`
- 6)
- 7)

✗ A co sme asi tak dostali?

PRIORITA CITATELOV

R():

- 1)
- 2) `readLS.lock(noWriters)`
- 3) `// kod citatelov`
- 4) `readLS.unlock(noWriters)`
- 5)
- 6)
- 7)

W():

- 1)
- 2)
- 3) `noWriters.wait()`
- 4) `// kod zapis.`
- 5) `noWriters.signal()`
- 6)
- 7)

✘ Klasické riešenie R-W s prioritou pre r()

PRIORITA CITATELOV

R():

- 1)
- 2) `readLS.lock(noWriters)`
- 3) `// kod citatelov`
- 4) `readLS.unlock(noWriters)`
- 5)
- 6)
- 7)

W():

- 1)
- 2)
- 3) `noWriters.wait()`
- 4) `// kod zapis.`
- 5) `noWriters.signal()`
- 6)
- 7)

- ✗ Klasicke riesenie R-W s prioritou pre r()
- ✗ Kde mame moznost starvation pre w()

POUCENIE

✗ Problem vylucenia kategorii

POUCENIE

- ✗ Problem vylucenia kategorii
 - + Extremne citlivy na model

POUCENIE

- ✗ Problem vylucenia kategorii
 - + Extremne citlivy na model
 - + Podla modelu riesime problem vyhladovenia

POUCENIE

- ✗ Problem vylucenia kategorii
 - + Extremne citlivy na model
 - + Podla modelu riesime problem vyhladovenia
 - + Podla modelu riesime zvysenie priority

POUCENIE

- ✗ Problem vylucenia kategorii
 - + Extremne citlivy na model
 - + Podla modelu riesime problem vyhladovenia
 - + Podla modelu riesime zvysenie priority
 - + V zakladnej verzii ta kategoria, ktora pouziva vypinac, moze byt zvyhodnena oproti tej, kde sa vypinac nepouziva!