

Referát z OS

Keywords z cvičenia 7

semafor

Je abstraktný datový typ a zároveň aj všeobecný synchronizačný prostriedok. Potrebuje podporu v jadre OS. Jeho základné

operácie sú init, wait, signal. Wait a Signal sú zároveň v kritickej oblasti semaforu. Hodnota semaforu určuje počet

operácií wait, ktoré sa môžu vykonať bez čakania. Nepoužíva obsadzujúce čakanie.

operácie nad semaforom

init - nastaví počiatočnú hodnotu semaforu, ktorá predstavuje, že koľko funkcií wait sa môže vykonať bez čakania

wait - zavola sa pred spustením procesu, ktorý volá wait

semafor obsahuje celé nezáporné číslo a ak to číslo je väčšie ako 0, tak sa toto číslo dekrementuje a wait skončí

a daný proces môže vstúpiť do KO a vykonávať sa.

Ak je toto číslo rovné nule, tak sa proces uspi a zaradi sa do radu čakajúcich procesov

signal - zavola sa po skončení procesu, keď vystúpi z KO

pozrie sa na rad procesov, ak je neprázdny, tak sa vyberie jeden z uspatých procesov (ze ktorý, to záleží od implementácie)

a zobudí ho. Ak je rad čakajúcich procesov prázdny, tak inkrementuje hodnotu semaforu

sem_value: 1

P1 wait // dekrementuje na 0

P2 wait // dá sa do queue

P1 work

P1 signal // vybere P2

P2 work

P2 signal // inkrementuje na 1

invariant semaforu

oznacme nw ako pocet dokoncenych operacii wait, ns - ako pocet dokoncenych operacii signal, c ako pociatocnu hodnotu semaforu,

potom plati nasledovny vzťah:

$nw(sem) \leq ns(sem) + c(sem)$

vseobecny semafor

nadobuda hodnoty nezaporne cisla

binarny semafor

nadobuda len hodnoty 0 a 1 - znaci sa tiez mutex akoze mutual exclusion, sluzi na riesenie vzajomneho vylucovania

inicialna hodnota semaforu

predstavuje počet nejakeho prostriedku, kde ten prostriedok je KO a moze ho vyuzivat iba max. jeden proces. Ze mam recepčnu

na intrakoch a mam 200 klučov od 200 izieb. Kľuč od izby je KO, moze ho mat len jeden proces, ale mam 200 takych klučov a to

je ta hodnota semaforu.

monitor

Vychadza sa z konceptu triedy v zmysle OOP. Je potrebna podpora kompilatora. Monitor je udajovy typ a združuje premenne

a operacie nad nimi. Premenne su privatne, zvonku neprístupne. Operacie su verejne. Procesy vstupuju do monitora volanim

jeho operácii. Operácie sa v čase vylučujú.

Sed

- Slúži na spracovanie textu, ktorý cez neho prúdi.
- Je silný nástroj na transformáciu a úpravu dát, vhodný na naučenie.

Princíp činnosti:

1. Prečíta riadok
2. Vykoná príkazy
3. Vypíše riadok

Syntax

sed OPTIONS... [SCRIPT] [INPUTFILE]...

Príkazy a Adresy

- p: vypísanie riadku
- =: výpis čísla riadku
- n: prejdí na ďalší riadok
- i: vloženie textu pred riadok
- a: vloženie textu za riadok
- c: vloženie textu namiesto riadka
- d: vymazanie riadku
- s/vzorka/nahrada/[flags]: substitúcia, nahrádzanie

Úlohy

Vypísať všetky riadky okrem siedmeho až desiateho.

```
sed '7,10d' list.txt
```

Zmeniť 'Zuzka' za 'Lucka', len v treťom riadku.

```
sed '3s/Zuzka/Lucka/g' list.txt
```

Zmeniť 'Zuzka' za 'Lucka' vo vete 'Zuzka, ja Ťa milujem.' (použiť poradie výskytu).

```
sed '/Zuzka, ja Ta milujem./s/Zuzka/Lucka/g' list.txt
```

Zmeniť 'Zuzka' za 'ZUZKA'.

```
sed 's/Zuzka/ZUZKA/g' list.txt
```

Zmeniť 'Zuzka' za 'zUZKA'.

```
sed 's/Zuzka/zUZKA/g' list.txt
```

Zmeniť 'ZUZka' za 'zuZKA'.

```
sed 's/ZUZka/zuZKA/g' list.txt
```

Zmeniť 'mozno' za 'nemozno'. 'Nemozno' nech nie je 'nenemozno'.

```
sed 's/mozno/nemozno/g; s/nenemozno/nemozno/g' list.txt
```

Vypísať iba báseň tak, že z nej odstrániť každý nepárny riadok.

```
sed '1d;n;d' list.txt
```

Vypísať iba báseň tak, že z nej odstrániť každý párný riadok.

```
sed 'n;d' list.txt
```

Prepísať v súbore 'list.txt' 'Zuzka' na 'Lucka'.

```
sed -i 's/Zuzka/Lucka/g' list.txt
```

```
sed 's/Zuzka/Lucka/g' list.txt
```

Vypísať iba báseň s číslami riadkov, ktoré budú na začiatku každého riadka.

```
sed '=' list.txt | sed 'N;s/\n/ /'
```

Bonus

```
#!/bin/bash
```

```
awk '{print $1}' access.log | sort | uniq > original_ips.txt
```

```
anonym_ip=100
```

```
awk '{print $1 " 123.45.67."anonym_ip++}' original_ips.txt > mapped_ips.txt
```

```
cp access.log anon.log
```

```
while IFS= read -r line; do
```

```
    original_ip=$(echo $line | awk '{print $1}')
```

```
    mapped_ip=$(echo $line | awk '{print $2}')
```

```
    sed -i "s/$original_ip/$mapped_ip/g" anon.log
```

```
done < mapped_ips.txt
```

```
awk '{print $1}' anon.log | sort | uniq -c | sort -nr | head
```