

Stiahnite si súbor `cv07_shareMem_mapMem_semaphore.tar.gz` a rozbaľte príkazom:
`tar -xvf cv07_shareMem_mapMem_semaphore.tar.gz`

Súbor obsahuje dva projekty zamerané na prácu so zdieľanou pamäťou, mapovanou pamäťou a semaformi. Obsah podadresárov:

- `common` obsahuje spoločné časti projektov
- `shareMem` obsahuje projekt zameraný na prácu so zdieľanou pamäťou a semaformi
- `mapMem` obsahuje projekt zameraný na prácu s mapovanou pamäťou a semaformi

Poznámka: pre prácu zo zdieľanou a mapovanou pamäťou sú užitočné príkazy `ipcs` a `ipcrm` zadané konzole.

1 zadanie (podadresár `shareMem`)

Zadanie je zamerané na použitie zdieľanej pamäte a semaforov. Rodičovský proces spustí detské procesy reprezentujúce učiteľa a študenta. Učiteľ zadá domácu úlohu študentovi, študent ju vypracuje a odovzdá učiteľovi. Domáca úloha bude zadaná a vypracovaná na hárku papiera, ktorý bude reprezentovaný zdieľanou pamäťou. Procesy učiteľa a študenta sa budú synchronizovať pomocou semaforov. Postup zadania, vypracovania a odovzdania domácej úlohy:

- učiteľ zadá domácu úlohu na hárku papiera (zapíše do zdieľanej pamäte)
- učiteľ oznámi zadanie domácej úlohy študentovi (pomocou semaforu)
- po oznámení študent vypracuje domácu úlohu (prejde cez semafor, zapíše vypracovanie do zdieľanej pamäte)
- študent oznámi učiteľovi vypracovanie úlohy (pomocou semaforu)
- učiteľ prečíta vypracovanú úlohu (prejde cez semafor, prečíta obsah zdieľanej pamäte)

Okrem toho budú procesy vypisovať kontrolné informácie.

Úloha bude zameraná na sčítavanie celých čísiel. Učiteľ zadá dve čísla, úlohou študenta bude vypočítať ich súčet. V zdieľanej pamäti musí byť priestor minimálne pre 3 čísla typu `int`. Prvé dve pre zadanie sčítancov, tretie pre ich súčet. Výpočet adresy týchto čísiel je možné realizovať pripočítaním násobku `sizeof(int)` ku adrese začiatku zdieľanej pamäte, ale takáto implementácia by bola neprehľadná. Preto s obsahom zdieľanej pamäte budeme pracovať ako so štruktúrou `Homework` (definovaná v spoločnej časti). Táto štruktúra obsahuje 3 čísla. Veľkosť segmentu zdieľanej pamäte musí byť minimálne `sizeof(Homework)`.

Pre synchronizáciu je potrebné použiť 2 semaforey. Jeden pre synchronizáciu zadania domácej úlohy (učiteľom) a čakania na zadanie úlohy (študentom). Druhý na synchronizáciu odovzdania vypracovanej úlohy (študentom) a čakania na vypracovanú úlohu (učiteľom).

Pseudokód inicializácie a procesov učiteľa a študenta je v zdrojovom súbore `shareMem.c`.

Postupne vypracujte nasledujúce úlohy

- a) Do `main-u` pred `fork` doplňte vytvorenie zdieľanej pamäte (`shmget`). Ako kľúč zdieľanej pamäte použite `IPC_PRIVATE`. Id zdieľanej pamäte uložte do premennej `shmId`.
- b) Do `main-u` pred `fork` doplňte vytvorenie množiny dvoch semaforov (`semget`). Ako kľúč množiny semaforov použite `IPC_PRIVATE`. Id množiny semaforov uložte do premennej `semId`.
- c) Do `main-u` pred `fork` doplňte inicializáciu hodnôt semaforov na nuly (`semctl`). Najprv ale podľa manuálu (`man semctl`) doplňte definíciu union-u, ktorú budete potrebovať pre zadanie 4. parametra funkcie `semctl`.
- d) Do procesu učiteľa (funkcia `RunTeacherProcess`) doplňte pripojenie zdieľanej pamäte (`shmat`). Adresa začiatku zdieľanej pamäte bude v premennej `homework`.
- e) Do procesu študenta (funkcia `RunStudentProcess`) doplňte pripojenie zdieľanej pamäte (`shmat`). Adresa začiatku zdieľanej pamäte bude v premennej `homework`.
- f) Do procesu učiteľa (za vygenerovanie a výpis domácej úlohy) doplňte oznámenie o zadaní domácej úlohy (vykonajte operáciu `post` (signal) nad semaforom `SEM_HW_ASSIGNED`) (`semop`).
- g) Do procesu študenta doplňte čakanie na zadanie domácej úlohy (vykonajte operáciu `wait` nad semaforom `SEM_HW_ASSIGNED`) (`semop`).
- h) Otestujte program (synchronizáciu zadania domácej úlohy). Kontrolné výpisy sú už naprogramované. Program v systéme necháva vytvorené segmenty zdieľanej pamäte a množiny semaforov. Tieto vytvorené systémové prostriedky môžete vypísať príkazom `ipcs` zadaným na konzole. Odstráňte ich zo systému príkazom `ipcrm` zadaným na konzole.
- i) Do procesu študenta doplňte oznámenie o vypracovaní úlohy (vykonajte operáciu `post` (signal) nad semaforom `SEM_HW_COMPLETE`) (`semop`).
- j) Do procesu učiteľa doplňte čakanie na vypracovanie úlohy (vykonajte operáciu `wait` nad semaforom `SEM_HW_COMPLETE`) (`semop`).
- k) Otestujte program (synchronizáciu).
- l) Do procesu učiteľa doplňte odpojenie zdieľanej pamäte (`shmdt`).
- m) Do procesu študenta doplňte odpojenie zdieľanej pamäte (`shmdt`).
- n) Do rodičovského procesu (za čakanie na dokončenie detských procesov) doplňte označenie zdieľanej pamäti na odstránenie zo systému (`shmctl, IPC_RMID`).
- o) Do rodičovského procesu doplňte odstránenie množiny semaforov.
- p) Ak v systéme zostali vami vytvorené segmenty zdieľanej pamäte a množiny semaforov, tak ich vymažte príkazom `ipcrm` zadaným z konzoly.
- q) Z manuálovej stránky (`man shmat`) zistite, či pripojené segmenty zdieľanej pamäte zostávajú pripojené po volaní `fork`, `execve` a `_exit`.
- r) Kedy je potrebné a kedy je chybou použitie `SEM_UNDO` (`semop`)?

2 zadanie (podadresar mapMem)

Toto zadanie je podobné predchádzajúcemu. Rozdiel je iba v tom, že namiesto zdieľanej pamäte je použitá mapovaná pamäť.

Postupne vypracujte nasledujúce úlohy

- a) Vo funkcii `main` pred volanie `fork` doplňte vytvorenie a otvorenie súboru s jedinečným názvom. Použite funkciu `mkstemp`. Súbor bude slúžiť na komunikáciu procesov. Deskriptor súboru uložte do premennej `fileDescr`.
- b) Zabezpečte, aby mal súbor potrebnú veľkosť (aby bola jeho veľkosť rovná veľkosti štruktúry `Homework` (`lseek, write`)).
- c) Vo funkcii `main` pred volanie `fork` doplňte vytvorenie množiny dvoch semaforov. Ako kľúč množiny semaforov použite `IPC_PRIVATE`. Id množiny semaforov uložte do premennej `semId`.
- d) Vo funkcii `main` pred volanie `fork` doplňte inicializáciu hodnôt semaforov na nuly. Najprv ale podľa manuálu (`man semctl`) doplňte definíciu union-u, ktorú budete potrebovať pre zadanie 4. parametra funkcie `semctl`.
- e) Vo funkcii `main` doplňte (za vytvorenie detských procesov, ale pred čakanie na ukončenie detských procesov) zatvorenie súboru a jeho odstránenie zo súborového systému (`close, unlink`).
- f) Vo funkcii `main` doplňte (za čakanie na ukončenie detských procesov) odstránenie množiny semaforov (`semctl`).
- g) V procese učiteľa (funkcia `RunTeacherProcess`) doplňte namapovanie súboru do pamäte (`mmap`). Adresa začiatku mapovanej pamäte bude v premennej `homework`.
- h) V procese študenta (funkcia `RunStudentProcess`) doplňte namapovanie súboru do pamäte (`mmap`). Adresa začiatku mapovanej pamäte bude v premennej `homework`.
- i) Do procesu učiteľa (za namapovanie súboru) doplňte zatvorenie súboru (`close`).
- j) Do procesu študenta (za namapovanie súboru) doplňte zatvorenie súboru (`close`).
- k) Do procesu učiteľa doplňte oznámenie o zadaní domácej úlohy (vykonajte operáciu `post (signal)` nad semaforom `SEM_HW_ASSIGNED`) (`semop`).
- l) Do procesu študenta doplňte čakanie na zadanie domácej úlohy (vykonajte operáciu `wait` nad semaforom `SEM_HW_ASSIGNED`) (`semop`).
- m) Otestujte program (synchronizáciu zadania domácej úlohy). Kontrolné výpisy sú už naprogramované.
- n) Do procesu študenta doplňte oznámenie o vypracovaní úlohy (vykonajte operáciu `post (signal)` nad semaforom `SEM_HW_COMPLETE`) (`semop`).
- o) Do procesu učiteľa doplňte čakanie na vypracovanie úlohy (vykonajte operáciu `wait` nad semaforom `SEM_HW_COMPLETE`) (`semop`).
- p) Do procesu učiteľa doplňte odmapovanie suboru z pamäte (`munmap`).
- q) Do procesu študenta doplňte odmapovanie suboru z pamäte (`munmap`).
- r) Otestujte, či program funguje aj bez nastavovania minimálnej potrebnej veľkosti, ktoré ste vykonali v bode b).