

Úvod

Stiahnite si súbor `cv05_06.tar.gz` a rozbaľte

stiahnutie: `wget http://aladin.elf.stuba.sk/~novak/sp/cv05_06/cv05_06.tar.gz`

rozbalenie: `tar -xvf cv05_06.tar.gz`

V tomto súbore sú 4 projekty. Každý je vo zvlášť adresári, spoločná časť je v adresári `common`. Projekty týkajúce sa nepomenovaných a pomenovaných rúr (adresáre `pipe`, `fifo`) sú určené na 5. cvičenie, projekty týkajúce sa front správ a soketov (adresáre `msg_queue`, `socket`) sú určené na 6. cvičenie.

Projekty obsahujú viacero zdrojových súborov. Na zostavenie programu z viacerých súborov je potrebné najprv spustiť kompilátor niekoľko krát, potom spustiť linker. Pre automatizované spúšťanie kompilátora a linkera slúži program `make`. Program `make` potrebuje informácie o projekte, podľa ktorých bude zostavovať program. Tieto informácie sú zapísané v súbore, ktorý sa väčšinou nazýva `makefile`, alebo `Makefile`. Ak programu `make` nešpecifikujeme názov súboru, automaticky hľadá súbor s názvom `makefile`, alebo `Makefile` v aktuálnom adresári. Preto pre zostavenie programu väčšinou stačí zadať príkaz `make` bez parametrov (čo bude platiť aj v týchto projektoch).

Pri práci na projektoch vystačíme len s dvoma možnosťami spúšťania `make`:

- zostavenie programu zo zdrojových súborov
- vymazanie súborov ktoré boli vygenerované pri zostavovaní programu (objektové súbory s príponou `.o` a spustiteľné súbory)

Pre zostavenie programu stačí prejsť do jedného z podadresárov (`pipe`, `fifo`, `msg_queue`, `socket`) a zadať príkaz `make`. Pre vymazanie vygenerovaných súborov stačí zadať príkazu `make` parameter `clean` (zadať `make clean`).

Zadania sú zamerané na medziprocesorovú komunikáciu. Procesy budú pomocou rôznych mechanizmov prenášať údaje o študentoch. V spoločnej časti pre všetky projekty (podadresár `common`) je definovaná štruktúra `STUDENT` reprezentujúca informácie o študentovi. Ďalej sú definované funkcie pre zjednodušenie nastavenia položiek štruktúry `STUDENT` a výpis informácii o študentovi na štandardný výstup. Ďalej je v spoločnej časti definované makro, ktoré je možné použiť pre kontrolovanie návratových hodnôt funkcií nastavujúcich premennú `errno`. Makro vypíše popis chyby (podľa premennej `errno`) a ukončí program.

1. zadanie (podadresár `pipe`)

Zadanie je zamerané na komunikáciu procesov pomocou nepomenovanej rúry. Rodičovský proces (implementovaný funkciou `runWriter`) bude posilať údaje o študentoch detskému procesu (funkcia `runReader`).

- Vytvorte nepomenovanú rúru pred volaním `fork`.
- Nastavte parametre volaní funkcií `runReader` a `runWriter` tak, aby prvý parameter bol koniec rúry určený na čítanie, druhý parameter koniec rúry určený na zápis.
- Na začiatku funkcií `runReader` a `runWriter` zatvorte konce rúry, ktoré nebudú procesy potrebovať.
- Do funkcie `runWriter` doplňte poslanie informácii aspoň o dvoch študentov
- Na koniec funkcie `runWriter` doplňte zatvorenie zapisovacieho konca rúry.
- Do funkcie `runReader` doplňte čítanie informácii o študentoch z rúry a výpis prijatých údajov na štandardný výstup. Proces ale nesmie mať napevno nastavený počet študentov, o ktorých ma prijať informácie. Musí automaticky detekovať, kedy už neprídu ďalšie údaje.
- Na koniec funkcie `runReader` doplňte zatvorenie konca rúry určeného na čítanie.
- Program by mal byť funkčný. Otestujte čo sa stane, keď v jednom z procesov zakomentujete zatvorenie konca rúry určeného na zápis.

2. zadanie (podadresár `fifo`)

Zadanie je zamerané na komunikáciu procesov pomocou pomenovanej rúry. Výstupom budú 2 programy, jeden pre posielanie údajov, druhý pre čítanie údajov.

V súbore `commonFifo.h` je definované makro obsahujúce názov pomenovanej rúry. Cesta ku pomenovanej rúre je relatívna, preto obidva programy musia byť spúšťané z toho istého adresára.

- Programy musia fungovať dobre bez ohľadu na to, ktorý z nich bude spustený prvý. Preto na začiatok obidvoch programov doplňte vytvorenie pomenovanej rúry. Ak volanie funkcie `mkfifo` skončí s chybou z dôvodu, že rúra už existuje, program musí pokračovať ďalej (chyba `EEXIST`).
- Do obidvoch programov doplňte otvorenie rúry na zápis, alebo čítanie. Otestujte či programy prejdú volanie `open` okamžite. Kedy programy prejdú volanie `open` (`man 3 mkfifo`, alebo `man 7 fifo`)?
- Do jedného z programov doplňte za `open`, odstránenie pomenovanej rúry zo súborového systému (obidva programy budú môcť rúru používať až do jej uzatvorenia).
- Do `fifoSender.c` doplňte poslanie informácii aspoň o 2 študentoch a zatvorenie rúry.
- Do `fifoReceiver.c` doplňte čítanie informácii o študentoch z rúry a výpis informácii na štandardný výstup. Doplňte aj zatvorenie rúry.