

Semestrální úloha X36OSY 2007

Téma úlohy: Producenti a konzumenti

Vypracoval: Tomas Horacek

6. ročník, obor Výpočetní technika, K336 FEL ČVUT,
Karlovo nám. 13, 121 35 Praha 2

Zadání úlohy

Mějme m producentů a n konzumentů. Každý producent má svojí frontu o kapacitě k položek, do které v náhodných intervalech zapisuje pokud je v ní volno, jinak se zablokuje a čeká. Položky ve frontách obsahují číslo producenta a pořadové číslo, ve kterém byly zapsány do fronty. Konzumenti v náhodných intervalech čtou položky z front. Každý konzument musí přečíst všechny položky od všech producentů, tzn. že položky se mohou odstranit z front až po přečtení všemi konzumenty.

Vstup: Producenti a konzumenti jsou reprezentováni vlákny/procesy a náhodně generují/čtou položky.

Výstup: Informace o tom, jak se mění obsah jednotlivých front a informace v jakém stavu se producenti a konzument nacházejí.

Analýza úlohy

Řešení úlohy pomocí vláken

Resení je za pomoci mutexu, pseudokod popisující algoritmus je zde:

```
def my_producent():
    for cislo_prvku in range(LIMIT_PRVKU):
        while je_fronta_plna():
            pockej_na_uvolneni_prvku()

        zamkni_frontu()
        pridej_prvek_do_fronty()
        odemkni_frontu()

        pockej_nahodnou_dobu()

    ukonci_vlakno()

def konzument():
    while True:
        pockej_nahodnou_dobu()

        ukonci_cteni = True
        for producent in range(M_PRODUCENTU):
            zamkni_frontu()
            if not fronta_je_prazdna():
                prvek = fronta[producent].front()

                if prvek.poradove_cislo != poradova_cisla_poslednich_prectenych_prveku[producent]:
                    prvek.pocet_precteni++

                if prvek.pocet_precteni == N_KONZUMENTU:
                    fronta[producent].pop()
                    delete prvek;
                    zavolej_uvolneni_prvku()

            poradova_cisla_poslednich_prectenych_prveku[producent] = prvek.poradove_cislo
            odemkni_frontu()

        if ukonci_cteni:
            ukonci_cteni = poradova_cisla_poslednich_prectenych_prveku[producent] == (LIMIT_PRVKU - 1)

    if ukonci_cteni:
        ukonci_vlakno()
```

Řešení úlohy pomocí procesů

Řešení je implementována jako $N + M + 1$ procesů. První proces vytvory producenty a konzumenty a ceka na jejich ukonceni. Fronty producentů (sdílená data) jsou uloženy v jednom segmentu sdílené paměti a jsou přístupné po volání `child_init()` přes ukazatel na pole `fronty`. Přístup ke frontám je řízen dvěma sadami semaforu (každá sada má velikost M), první sada urcuje, zda se da s frontou manipulovat, druhá, zda se dá do fronty přidávat. Sdílené prostředky alokuje a dealokuje první proces.

Signály SIGQUIT, SIGINT, SIGHUP a SIGTERM jsou odchyceny a dealokace sdílených prostředků probíhá správně i při jejich vyvolání.

Závěr

Uloha mi pomohla prakticky vyzkouset tvoreni vlaken, jejich synchronizaci pomoci vlaken a take vytvoreni procesu, sdilene pameti a obsluhu semaforu.

Literatura

1. Slides ze cviceni
2. Prikklady ze cviceni
3. manualove stranky
4. [The GNU C Library - Signal Handling](#)
5. [queue::queue - C++ Reference](#)