

Oblast vežbi : *Konkurentno programiranje*

PRIMER UPOTREBE VREMENSKIH KONTROLA U KONKURENTNOM PROGRAMIRANJU

Uvod

U ovoj vežbi primer proizvođač-potrošač je realizovan korišćenjem vremenskih kontrola. Proizvođač je realizovan kao funkcija koja se periodično poziva i ispituje da li pritisnut taster na tastaturi. Ukoliko jeste, čita se znak sa tastature i unosi u kružni bafer. Potrošač je realizovan kao programska nit koja čita znake iz kružnog bafera. Ukoliko je kružni bafer prazan određeni vremenski period program završava sa radom. Pored programske niti potrošača u primeru se formira i programska nit za ispisivanje sadržaja kružnog bafera koja je vremenski kontrolisana.

Proizvođač

Proizvođač je realizovan pomoću vremenske kontrole *timeSetEvent*.

```
MMRESULT timeSetEvent(  UINT          uDelay,
                        UINT          uResolution,
                        LPTIMECALLBACK lpTimeProc,
                        DWORD_PTR      dwUser,
                        UINT          fuEvent );
```

Funkcija	Opis
<i>timeSetEvent</i>	Stvara vremensku kontrolu
Parametri	Opis
UINT <i>uDelay</i>	Period vremenske kontrole u milisekundama.
UINT <i>uResolution</i>	Rezolucija vremenske kontrole u milisekundama (ms). Vrednost 0 označava formiranje vremenske kontrole sa najvećom mogućom tačnošću za dati sistem.
LPTIMECALLBACK <i>lpTimeProc</i>	Adresa funkcije koja će biti izvršena po isteku vremenske kontrole.
DWORD <i>dwUser</i>	Parametar koji će biti prosleđen funkciji po isteku vremenske kontrole.
UINT <i>fuEvent</i>	Definiše tip vremenske kontrole. Moguće su sledeće vrednosti ovog parametra: TIME_ONESHOT - Vremenska kontrola će biti pokrenuta samo jednom. TIME_PERIODIC - Vremenska kontrola će biti startovana periodično.

Povratna vrednost	Opis
<i>Promenljiva tipa MMRESULT</i>	Ako je poziv uspešan, povratna vrednost je ID vremenske kontrole koja se stvara. U slučaju greške vraća se NULL.

Funkcija *ProducerTimerFunc* se poziva periodično svakih 100ms i proverava da li je pritisnut taster na tastaturi sa funkcijom *kbhit()*.

```
void CALLBACK TimeProc( UINT uID,  
                        UINT uMsg,  
                        DWORD dwUser,  
                        DWORD dw1,  
                        DWORD dw2 );
```

Funkcija	Opis
<i>TimeProc</i>	Funkcija koja će biti izvršena po isteku vremenske kontrole. Definiše se od strane korisnika vremenske kontrole.
Parametri	Opis
UINT <i>uID</i>	Identifikator vremenske kontrole.
UINT <i>uMsg</i>	Rezervisano za kasniju upotrebu (ne koristi se).
DWORD <i>dwUser</i>	Parametar prosleđen pri stvaranju vremenske kontrole.
DWORD <i>dw1</i>	Rezervisano za kasniju upotrebu (ne koristi se).
DWORD <i>dw</i>	Rezervisano za kasniju upotrebu (ne koristi se).

Ukoliko je pritisnut taster, preuzima znak funkcijom *getch()* i pokušava da unese znak u deljeni resurs (kružni bafer). Postupak unošenja podatka u deljeni resurs je opisan u vežbi broj dva.

Ukoliko je pritisnuti znak 'q' proizvođač prekida vremensku kontrolu funkcijom *timeKillEvent*.

```
MMRESULT timeKillEvent( UINT uTimerID );
```

Funkcija	Opis
<i>timeKillEvent</i>	Zaustavljanje vremenske kontrole
Parametri	Opis
<i>uTimerID</i>	ID vremenske kontrole koja se zaustavlja
Povratna vrednost	Opis
<i>Promenljiva tipa BOOL</i>	Ako je poziv uspešan, povratna vrednost je TIMERR_NOERROR. Ako vremenska kontrola koja se zaustavlja ne postoji, povratna vrednost je MMSYSERR_INVALIDPARAM.

```
void CALLBACK ProducerTimerFunc(UINT timerID,UINT msg,DWORD dwUser,DWORD dw1,DWORD dw2)
{
    char c;
    //Ukoliko je taster na tastaturi pritisnut,
    //funkcija kbhit() vraca broj razlicit od nule.

    if(kbhit()){
        c = getch();
        // P(Empty)- operacija smanjuje semafor Empty za jedan
        // ukoliko je to moguće ; Ako je Empty=0
        // nit ceka RealeseSemaphore(Empty)
        WaitForSingleObject(Empty, INFINITE);

        // Ulazak u kritičnu sekciju BufferAccess kako bi se obezbedio
        // sinhronizovan pristup deljenoj promenljivoj (kružnom baferu ring)
        EnterCriticalSection(&BufferAccess);

        ringBufPutChar(&ring, c);

        // Napustanje kritične sekcije
        LeaveCriticalSection(&BufferAccess);

        // V(Full) - operacija uvecava semafor Full za jedan
        ReleaseSemaphore(Full,1,NULL);
        // Ukoliko je pritisnut taster 'q' prekida se
        // vremenska kontrola
        if (c == 'q'){
            timeKillEvent(hProduceTimer);
        }
    }
}
```

Potrošač

Programska nit potrošač, čeka na semaforu Full ako je njegova vrednost jednaka nuli (bafer prazan). Ukoliko je kružni bafer za vremenski period od 5000ms (5 sekundi) prazan, funkcija *WaitForSingleObject* kao rezultat vraća signal isteka vremenske kontrole i program završava sa radom.

Ukoliko bafer nije prazan (Full različito od nule), potrošač pokušava da uđe u kritičnu sekciju *BufferAccess*, pomoću funkcije *EnterCriticalSection*. Ukoliko

proizvođač nije prethodno zauzeo kritičnu sekciju BufferAccess, potrošač je zauzima i čita podatak iz bafera i ispisuje ga na konzolu. Nakon čitanja podatka iz bafera potrošač povećava vrednost semafora Empty za jedan.

```
DWORD WaitForMultipleObjects( DWORD nCount,
                              const HANDLE* lpHandles,
                              BOOL bWaitAll,
                              DWORD dwMilliseconds );
```

Funkcija	Opis
<i>WaitForMultipleObjects</i>	Čekanje na ostvarenje nekog od događaja. U slučaju semafora čekanje na pozitivnu vrednost nekog semafora ili svih semafora u nizu na koji pokazuje <i>lpHandles</i> . Funkcija vraća vrednosti od WAIT_OBJECT_0 do (WAIT_OBJECT_0 + <i>nCount</i> - 1). Ukoliko je <i>fWaitAll</i> FALSE povratna vrednost funkcije smanjena za WAIT_OBJECT_0 ukazuje na indeks objekta (semafora) u nizu koji signaliziran.
Parametri	Opis
<i>nCount</i>	Broj objekata u nizu na koji pokazuje <i>lpHandles</i>
<i>lpHandles</i>	Pokazivač na niz signalnih objekata (najčešće semafori).
<i>fWaitAll</i>	Specificira tip čekanja: <ul style="list-style-type: none"> - TRUE – čeka se na pozitivnu vrednost svih objekata u nizu - FALSE – čeka se na pozitivnu vrednost bilo kog objekta u nizu
<i>dwMilliseconds</i>	Vreme koje se čeka na semaforu u mili sekundama. Ukoliko se čeka neograničeno prosleđuje se konstanta INFINITE.

Ukoliko je pročitani znak iz bafera znak 'q' programska nit potrošač signalizira programskoj niti za ispis sadržaja kružnog bafera kraj rada funkcijom *ReleaseSemaphore* i završava sa radom.

```
DWORD WINAPI consumer(LPVOID param) {
    char c;
    DWORD result;
    while(1){

        // P(Full) - operacija smanjuje semafor Full
        // za jedan ukoliko je to moguće;
        // Ako je Full=0 nit ceka ReleaseSemaphore(Full);
        // Ukoliko je kružni bafer prazan 5 sekundi, funkcija
        // WaitForSingleObject vraća kao rezultat istek vremenske kontrole
```

```
// WAIT_TIMEOUT i završava se sa radom programa
result = WaitForSingleObject(Full, 5000);
switch(result){
case WAIT_OBJECT_0:

    // Ulazak u kritičnu sekciju BufferAccess
    EnterCriticalSection(&BufferAccess);

    // Citanje kruznog bafera;
    c = ringBufGetChar(&ring);

    // Napuštanje kritične sekcije;
    LeaveCriticalSection(&BufferAccess);

    // Ispis na konzolu;
    printf("Znak preuzet iz kruznog bafera je '%c' \n", c);

    // V(Empty) - operacija uvećava semafor Empty za jedan;
    ReleaseSemaphore(Empty,1,NULL);

    // Ukoliko je pritisnut taster 'q' programska nit potrošač:
    // - signalizira programskoj niti za ispis sadržaja kruznog
    //   bafera kraj rada operacijom V(FinishPrint)
    // - završava sa radom
    if(c == 'q'){
        ReleaseSemaphore(FinishPrint,1,NULL);
        return 0;
    }
    break;

case WAIT_TIMEOUT:
    printf("\nKružni bafer je prazan 5 sekundi \n");
    printf("Kraj programa !!! \n");
    exit(0);

}

Sleep(SLEEPING_TIME);

}
return 0;
}
```

Programska nit za ispis sadržaja kružnog bafera

Programska nit za ispis sadržaja bafera čeka na pozitivnu vrednost brojača semafora StartPrint ili brojača semafora FinishPrint. Brojač semafora StartPrint se inkrementira pomoću vremenske kontrole.

```
void CALLBACK PrintStateTimerFunc(UINT timerID,UINT msg,DWORD
dwUser,DWORD dw1,DWORD dw2)
{
    ReleaseSemaphore(StartPrint,1,NULL);
}
```

Naime, pomoću vremenske kontrole, periodično svakih 1000ms (1 sekund) se poziva funkcija *PrintStateTimerFunc* koja povećava vrednost brojača *StartPrint*. Kada je vrednost semafora *StartPrint* pozitivna, programska nit za ispis, ispisuje sadržaj kružnog bafera, vodeći računa o pristupu deljenom resursu.

Vrednost brojača semafora *FinishPrint* se povećava u programskoj niti potrošač, ukoliko je pročitani znak iz bafera 'q'. Ukoliko je vrednost semafora *FinishPrint* pozitivna, programska nit za ispis završava sa radom.

```
DWORD WINAPI PrintStateThreadFunc(LPVOID param) {

    HANDLE hSemaphores[2] = { StartPrint, FinishPrint};

    // Cekanje na pozitivnu vrednost brojaca semafora StartPrint
    // ili brojaca semafora FinishPrint;
    // Ukoliko je vrednost brojaca semafora StartPrint pozitivna,
    // ispisuje se sadrzaj bafera;
    // Ukoliko je vrednost brojaca semafora FinishPrint pozitivna,
    // programska nit za ispis završava sa radom;
    while(WaitForMultipleObjects(2, hSemaphores, FALSE,INFINITE) ==
WAIT_OBJECT_0)
    {
        // Ulazak u kritičnu sekciju BufferAccess
        EnterCriticalSection(&BufferAccess);

        printf("Sadržaj kruznog bafera: ");
        for(int i=0; i<RING_SIZE;i++)
            printf("%c",ring.data[i]);
        printf("\n");

        // Napustanje kritične sekcije;
        LeaveCriticalSection(&BufferAccess);
    }
    return 0;
}
```

Zadatak

1a). U postojećem primeru dodati vremensku kontrolu koja smanjuje globalnu promenljivu *counter* svake sekunde (*counter* je inicijalno na 5). Ukoliko je *counter* jednak 0 završava se sa radom programa. Promenljiva *counter* se postavlja na 5 svaki put kada potrošač pročita znak iz bafera. Na ovaj način se realizuje završetak programa ukoliko u bafer nije unet znak u periodu od 5 sekundi. Za završetak programa **NE** koristiti funkciju *exit*! Iz koda programske niti potrošač, izbaciti postojeću vremensku kontrolu (funkcija *WaitForSingleObject* treba da čeka beskonačno).

Prilikom realizacije voditi računa o sinhronizovanom pristupu deljenoj promenljivoj *counter*.

1b). U postojeći primer ubaciti podršku za uključivanje i isključivanje vremenske kontrole za završetak programa na sledeći način:

- Ako je stisnut taster 'f' isključiti vremensku kontrolu, program čeka neograničeno na sledeći znak
- Ako je stisnut taster 'n' uključiti vremensku kontrolu, program čeka 5 sec na sledeći znak.

Vremenska kontrola je uključena na početku.