

Есептеу процестердің теориясы

Дәріс 4

Құрастырушы: А. Ж. Сарина, аға оқытушы

# Ағындарды синхронизациялау

- Ағындар программаны өңдеуде және құрастыруда жұмысты жеңілдетеді, сонымен қатар жылдамдығын жоғарлатады.
- **Ағындарды синхронизациялаудың қажеттілігі**
- Көбінесе екі немесе одан көп ағындарды сол мезетте орындалып жатқан кезде координациялау үшін қажеттіліктер туады.
- Мысалы, бірнеше ағындар бір айнымалыға қатынау кезінде болады. Ал басқа жағдайларда бір ағын, басқа бір ағынның жұмысы аяқталмайынша, ол өзінің жұмысын жалғастыра алмайды. Міне, осындай жағдайларда синхронизациялау қажеттіліктері туады.

- **Ағындарды синхронизациялау объектілері**
- Ағындарды және үрдістерді синхронизациялау үшін арналған Win 32 төрт объектілерді ұсынады. Олардың үшеуі – мьютекстер, семафорлар мен істер – бұлар ядроның объектілері болып табылады және дескрипторлары болады. Төртінші объект - CRITICAL\_SECTION. CRITICAL\_SECTION объектілерінің ыңғайлылығына және жылдамдығының жоғары болуына байланысты, бұлардың механизмдері қажеттірек. Егерде программаның қажеттілігіне сәйкес болған жағдайда.

## ■ **CRITICAL\_SECTION** объектісі



Критикалық секция – бұл кодтың бөлігі, яғни бір мезетте тек ғана ағынмен орындалады. Ал мұндай секцияның бір мезетте бірнеше ағындармен орындалса, онда қате нәтижелерге алып келер еді.

CRITICAL\_SECTION типтерінің объектілері (критикалық секция - КС) орнатылуы және жойылуы мүмкін, бірақ олар басқа үрдістермен бөлінбейді және дескрипторлары болмайды. Ағындар критикалық секцияға кіреді және одан шығып кетеді. Критикалық секцияда бір ғана ағын болады. Бір ғана ағын критикалық секцияға кіріп, программаның бірнеше жерінен шығып кете алады.

- CRITICAL\_SECTION типтерінің айнымалысында инициализациялау және жою үшін мына функцияларды қолданады:
- VOID InitializeCriticalSection (LPCRITICAL\_SECTION lpCriticalSection)
- VOID DeleteCriticalSection (LPCRITICAL\_SECTION lpCriticalSection)

- EnterCriticalSection функциясы ағындарды бұғаттайды, егерде берілген секцияда басқа ағын болса. LeaveCriticalSection функциясы басқа ағын орындалып жатқан кезде, күтіп тұрған ағымды бұғаттау алынады. Критикалық секцияға басқаруды беріп тұру керек, егер бермесек, онда басқарушы ағын аяқталса да, басқа ағындар күту режимінде тұра берер еді.
- КС – ның бұғатталуы:  
VOID EnterCriticalSection (LPCRITICAL\_SECTION  
lpcsCriticalSection)  
VOID LeaveCriticalSection  
(LPCRITICAL\_SECTION lpcsCriticalSection)

- Егер ағын КС-н иеленсе, онда ол секцияға бұғаттаусыз қайтадан кіре алады. Критикалық секцияда есептеуіш болады, сондықтан ағын секцияға қанша рет кірсе, сонша рет секциядан шығу керек. Себебі, басқа ағындарға да критикалық секцияны босату керек. `EnterCriticalSection` функциясына күту уақыты шектелмейді, критикалық секцияны иеленген ағын секциядан шыққан кезде, осы мезетте басқа ағын бұғатталады. `TryEnterCriticalSection` функциясы арқылы КС басқа ағын иеленген немесе иеленбегенін анықтауға болады.

# Әрекет есептеуішін басқару

- BOOL TryEnterCriticalSection (LPCRITICAL\_SECTION lpCriticalSection)
- TRUE мәні қайтарылса, онда функцияны шақырған ағын КС иеленгенін білдіреді. Ал FALSE мәні қайтарылса, онда КС басқа ағын пайдаланып тұрғаннын білдіреді. CRITICAL\_SECTION артықшылығы, ол ядроның объектісі емес және қолданушылар аумағында орналасқан.
- EnterCriticalSection функциясы орындалуы кезінде ағын КС – ны бос емес екенін анықтаса, ол ядроға кіріп CRITICAL\_SECTION объектісі босағанша бұғатталады. Әрекет есептеуішін басқару үшін екі функция қолданады:
- SetCriticalSectionSpinCount бұл функция есептеуішті динамикалық түрде өзгертеді. InitializeCriticalSectionAndSpinCount функциясы InitializeCriticalSection функциясын алмастырады.
- CRITICAL\_SECTION объектісі синхронизациялау механизмі жоғары.