

5. Тәжірибелік сабақтарды өткізу жоспары және тапсырмаларды орындау бойынша әдістемелік нұсқаулықтар

Тәжірибелік сабақ көрсетілген тақырыптар бойынша білім алушылардың білік, дағдыларын қалыптастыруға бағытталады. Әрбір тәжірибелік сабақты орындап тапсыру үшін оқу материалдарын меңгеру, оқытушымен берілген тапсырмаларды орындау ұсынылады.

№ апталар	Тәжірибелік сабақтарды атауы	Сағат саны
1	<p>Параллель есептеулер туралы негізгі түсініктер.</p> <ol style="list-style-type: none"> 1. Фазалық параллель 2. Тапсырмалар графы немесе бөлу және жеңу 3. Өндіруші мен тұтынушы немесе газ құбыры 4. Ферма өңдеу немесе мастер мен жұмысшы <p>Оқыту формасы: Компьютерлік практикум</p>	2
2	<p>Параллельді жүйе деректерін өңдеу классификациясының негіздері.</p> <ol style="list-style-type: none"> 1. SISD(Single Instruction Single Data) басқару типі бойынша құрылған жүйе 2. SIMD(Single Instruction Multiple Data) басқару типі бойынша құрылған жүйе 3. MISD(Multiple Instruction Multiple Data) басқару типі бойынша құрылған жүйе 4. MIMD(Multiple Instruction Multiple Data) басқару типі бойынша құрылған жүйе <p>Оқыту формасы: Компьютерлік практикум</p>	2
3	<p>Параллель есептеулер архитектурасы.</p> <ol style="list-style-type: none"> 1. Үлестіруші жады бар көлемді-параллельді компьютерлер 2. Векторлы-конвейерлік компьютерлер 3. Ортақ жады бар компьютерлер 4. Ортақ жадылы, таратылған жадылы және векторлы-конвейерлік компьютерлер комбинациясы <p>Оқыту формасы: Компьютерлік практикум</p>	2
4	<p>Білім берудегі параллель есептеулер кластерін баптау негіздері және кластерде параллель есептеулер.</p> <ol style="list-style-type: none"> 1. Matlab ортасында параллель есептеулер 2. Matlab ортасында параллель есептеулер кластерін баптау негіздері <p>Оқыту формасы: Компьютерлік практикум</p>	2
5	<p>Параллель есептеулер алгоритмі.</p> <ol style="list-style-type: none"> 1. Алгоритмнің параллель формасы 2. Тіктөртбұрыштың ауданын табу алгоритмі 3. Сандардың қосындысын есептеу алгоритмдері <p>Оқыту формасы: Компьютерлік практикум</p>	2
6	<p>Параллель программалау модельдері</p> <ol style="list-style-type: none"> 1. Жасырын параллельділік 2. Ашық параллельділік <p>Оқыту формасы: Компьютерлік практикум</p>	2
7	<p>Параллель есептеулер тиімділігі.</p> <ol style="list-style-type: none"> 1. Орындалу уақыты (<i>Execution Time</i>) 2. Жылдамдату (<i>SpeedUp</i>) 3. Эфективтілік (<i>Efficiency</i>) 4. Есептеу құны (<i>Cost</i>) <p>Оқыту формасы: Компьютерлік практикум</p>	2
8	<p>RadStudio ортасында параллель есептеулер.</p> <ol style="list-style-type: none"> 1. RadStudio ортасында параллель есептеулер модульдері 2. RadStudio ортасында параллель есептеулер жүргізу <p>Оқыту формасы: Компьютерлік практикум</p>	2
9	<p>RadStudio ортасында кроссплатформалық қосымшалар құруда параллель есептеулер қолдану.</p> <ol style="list-style-type: none"> 1. RadStudio ортасында кроссплатформалық қосымшалар құру 2. Цифрлық білім беру ресурстарын құруда параллель есептеулер қолдану. <p>Оқыту формасы: Компьютерлік практикум</p>	2
10	<p>Параллельдеудегі проблемалар</p> <ol style="list-style-type: none"> 1. Құрылымдық 	2

	2. Жалпыға ортақ 3. Тасымалдау Оқыту формасы: Компьютерлік практикум	
11	Сандық әдістерді параллель есептеу. 1. Сызықтық теңдеулер жүйесін параллель есептеу. 2. Матрицаны-векторға жол бойынша бөліп көбейту Оқыту формасы: Компьютерлік практикум	2
12	Сандық әдістерді параллель есептеу. 1. Анықталған интегралды параллель есептеу 2. Параллель есептеу арқылы ауытқу шамаларын азайту Оқыту формасы: Компьютерлік практикум	2
13	Сандық әдістерді параллель есептеу. 1. Дифференциальдық теңдеулерді параллель есептеу. 2. Параллель есептеу арқылы ауытқу шамаларын азайту Оқыту формасы: Компьютерлік практикум	2
14	Оптимизация есептерін параллель есептеу. 1. Ең ара қашықтық есебін параллель есептеу. 2. Параллель есептеу арқылы ауытқу шамаларын азайту Оқыту формасы: Компьютерлік практикум	2
15	GPU (Graphics Processing Unit) қосымша есептеу қуаты ретінде пайдалану. 1. GPU да есептеу 2. GPU да параллель есептеу және салыстыру Оқыту формасы: Компьютерлік практикум	2
БАРЛЫҒЫ		30

Тақырып 1. Параллель есептеулер туралы негізгі түсініктер.

Тапсырма 1:

1. www.top500.org сайтынан 2019 жылғы әлемдегі ең жылдам 10 суперкомпьютерлердің тізімі мен оның сипаттамаларын кесте түрінде жазып, салыстырыңыз.

Тапсырма 2:

Параллель есептеу даму кезеңдеріне баяндама жасаңыз .

Өткізілу формасы: компьютерлік практикум

Ұсынылатын әдебиет: 1-2,6

Әдістемелік нұсқаулар: практикалық сабақтың тақырыбына қарасты лекциялық материалды қарастырып, түсініктердің байланысын анықтау керек және программалау ортасында жүзеге асыру керек.

Бақылау сұрақтары:

1. Жоғары өнімді есептеу техникаларының бағыттарын атаңыз.
2. Деректерді параллельді өңдеудің қандай принциптері бар?
3. Параллельдік дегеніміздің түсінігі қандай?
4. Конвейерлік дегеніміздің түсінігі неде?
5. Қандай белгілі векторлы-конвейерлік суперкомпьютерлер бар?

Әдебиеттер

1. Серік М., Бакиев М.Н. Параллель есептеулер (MatLab ортасы) Оқу құралы.- Астана, 2014. -93б.
2. Дүйсембиев Е.Е. Параллель есептеулер: оқу құралы -Алматы: Дәуір, 2011.-272 б
3. www.top500.org

Тақырып 2. Параллельді жүйе деректерін өңдеу классификациясының негіздері.

Тапсырма 1:

а) ортақ жадты компьютерлер туралы ядроларының санына байланысты мысал келтіріңіз.

ә) желіде ұйымдастырылған үлестіруші жады бар компьютерлер туралы ядроларын кіріктіріп схема сызыңыз.

Тапсырма 2:

а) SIMD басқару типі бойынша құрылған жүйенің схемасын сызыңыз. ә) MIMD басқару типі бойынша құрылған жүйенің схемасын сызыңыз.

Тапсырма 3:

а) параллельдеуге жататын есептерге мысал келтіріңіз ә) параллельдеуге келмейтін есептерге мысалдар келтіріңіз. **Өткізілу формасы:** семинарлық сабақ.

Ұсынылатын әдебиет: 1-3,6

Әдістемелік нұсқаулар: практикалық сабақтың тақырыбына қарасты лекциялық материалды қарастырып, түсініктердің байланысын анықтау керек және программалау ортасында жүзеге асыру керек.

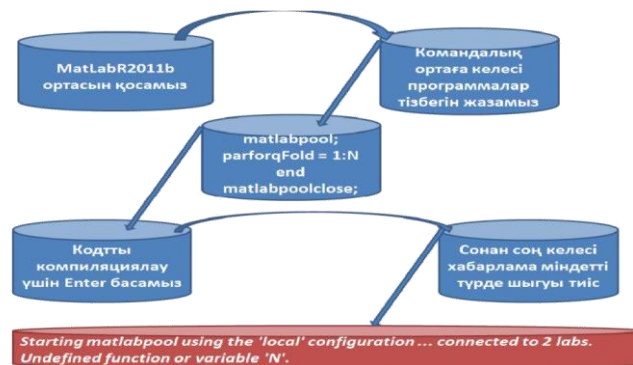
Тақырып 3. Параллель есептеулер архитектурасы.

Тапсырма 1:

Өз компьютерлік сыныбыңыздағы локальды желіні баптау керек.

Тапсырма 2:

Желіге қосылған компьютерлерді анықтау, мысалы келесі суретте көрсетілгендей және тағы қалай анықтауға болады.



Тапсырма 3:

Бірнеше компьютерге бірдей IP-адрес беріп, эксперимент жүзінде жұмыстың нәтижесін тексеру.

Өткізілу формасы: компьютерлік практикум

Ұсынылатын әдебиет: 1,2

Әдістемелік нұсқаулар: практикалық сабақтың тақырыбына қарасты лекциялық материалды қарастырып, түсініктердің байланысын анықтау керек және программалау ортасында жүзеге асыру керек.

Тақырып 4. Білім берудегі параллель есептеулер кластерін баптау негіздері

және кластерде

Тапсырма 1:

3 компьютерден тұратын кластерді құру кодын жазыңыз.

Тапсырма 2:

Құрастырған кластердің максималды және минималды жұмысшыларының санын есептеу керек (3 жұмысшыдан қосуға болады деген дерекпен).

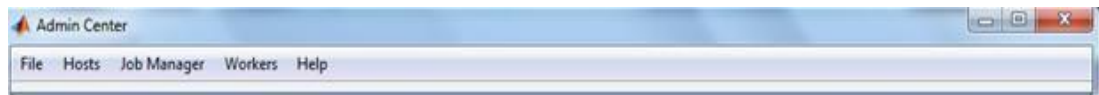
Әр компьютердің ядро саны 4- тең, ал әр ядродағы жіберілетін жұмысшы саны 3-ке тең

ұмытпай shift+enter батырма комбинацияларын басып отыру

```
D:\Documents and Settings\Administrator>arp -a
```

Тапсырма 3:

Әр серверге Admin Center арқылы қанша worker енгізу болатынын негіздеңіз.



Сурет - Admin Center терезесі

Өткізілу формасы: компьютерлік практикум

Ұсынылатын әдебиет: 1-3,5,6

Әдістемелік нұсқаулар: практикалық сабақтың тақырыбына қарасты лекциялық материалды қарастырып, түсініктердің байланысын анықтау керек және программалау ортасында жүзеге асыру керек.

Тақырып 5. Параллель есептеулер алгоритмі.

Тапсырма 1: Аудиториядағы компьютерлердің сипаттамасына талдау жасау.

Тапсырма 2: Компьютерлердің конвейерлік сипатын 4,5,6 деңгейінде талдау.

Тапсырма 3: Заманауи конвейерлердің сатылары туралы реферат дайындау.

Өткізілу формасы: компьютерлік практикум

Ұсынылатын әдебиет: 1-6

Әдістемелік нұсқаулар: практикалық сабақтың тақырыбына қарасты лекциялық материалды қарастырып, түсініктердің байланысын анықтау керек және программалау ортасында жүзеге асыру керек.

Тақырып 6. Параллель программалау модельдері

Тапсырма 1:

N тәуелсіз тапсырмалар жиынынан тұратын үлгіні жасап шығару:

1. MatLabR2011b аясын енгіземіз.

2. Командалық терезеге келесі кодты тереміз және кодтың әрбір тармағынан соң shift+enter батырма комбинацияларын басып отыру керек:

```
matlabpool; parforqFold = 1:N
```

```
%Any function could be executed inside parfor loop.
```

```
results{qFold} = DoCrossValidationIteration(qFold);
```

```
%The following output will be redirected to your
```

```
console: fprintf('parfor loop: task %d completed.\n', qFold);
end
matlabpoolclose;
```

3. Кодтың компиляциясынан кейін enter-ді басамыз. Келесі хабарлама пайда болу керек:

Starting matlabpool using the 'local' configuration ... connected to 2 labs.
Undefined function or variable 'N'.

Курсор тұрған жерден кейін есептеуге арналған нақты тапсырмаларды тереміз және

`matlabpool open local`

```
6 m1=1;
```

```
m2=3;
```

```
sum=0;
```

```
m4=rand(1,6);
```

```
parfor i = 1:6
```

```
  m1=i;
```

```
  sum=sum+i;
```

```
  m3(i)=m4(i);
```

```
  if i <= m2
```

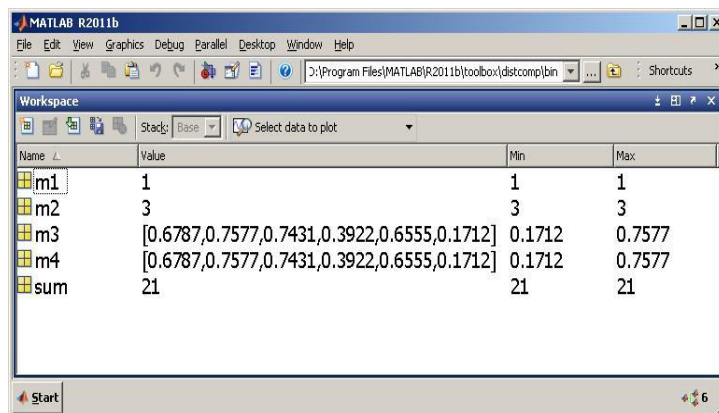
```
    m5=2*m1;
```

```
  end
```

```
end
```

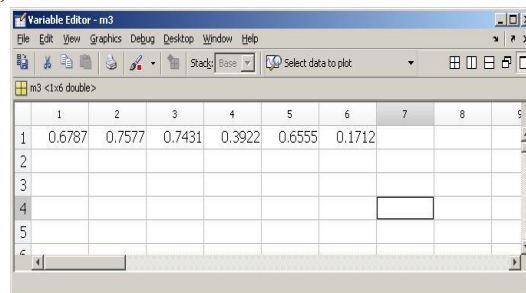
```
matlabpool close
```

Enter батырмасын басқаннан кейін Workspace терезесінде қорытынды шығады:



Сурет – нәтижесі бар Workspace терезесі

Аталған терезеде сәйкес айнымалының атын белгілей отырып және контекстік мәзірдің келесі командасын орындау арқылы есептеудің жауабын басқа форматта көруге болады: `openselection (ctrl+D)`, яғни :



Сурет – Жинақталған форматтағы есептеудің шешімі

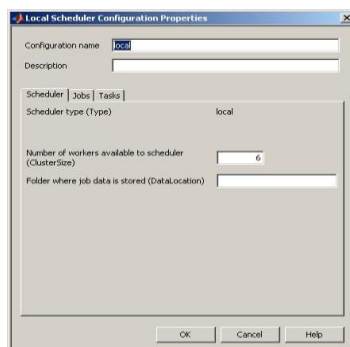
Сонымен, біз локальді компьютердің барлық ядросын қолдану үшін parfor операторының жұмысын жандандырдық.

Parfor заты бірнеше жекелеген matlab.exe үдерістерімен жүзеге асады.

Мәзірдің мына командасын орындай отырып:

Parallel – ManageConfigurations... - және пайда болған терезеде 'local' конфигурациясын таңдаймыз, содан соң контекстік мәзір көмегімен Properties... пунктін таңдағанда аламыз.

ClusterSize-де қалаулы мәнді енгізу мүмкіндігі бар (мысалы 6) LocalSchedulerConfigurationProperties терезесі пайда болады.



Сурет – Local Scheduler Configuration Properties терезесі

Аталған операция ClusterSize мәнінің ауысуына қарай түйін көппроцессорлы немесе көпядролы болған жағдайда ғана белгілі бір мәнге ие болады. 'local' конфигурациясының мәніне қарай ClusterSize мәні 4-ке тең болады.

Тапсырма 2:

Parfor операторын қолдану арқылы векторды екі бөлікке бөлу амалын орындау:

$$B(4) = \begin{matrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{matrix}$$

Өткізілу формасы: компьютерлік практикум

Ұсынылатын әдебиет: 1,2,6

Әдістемелік нұсқаулар: практикалық сабақтың тақырыбына қарасты лекциялық материалды қарастырып, түсініктердің байланысын анықтау керек және программалау ортасында жүзеге асыру керек.

Тақырып 7. Параллель есептеулер тиімділігі.

Мақсаты: RadStudio XE7 ортасында параллель есептеу үшін Threading модулының TTask класын қолдануды үйрену.

Есептің қойылуы: Жай есептеумен параллель есептеудің есептеу уақыттарын салыстыратын есептер қарастыру.

Жұмыс реті:

Есеп 1

Алдымен параллельді есептеудің тиімділігін көрсету үшін арнайы жай сандарды тізбектей есептейтін алгоритмді қарастырайық:

```
Function JaibasarMysal(a:integer):boolean;
```

```
//жай сан бола ма, жок па екендігін тексеретін функциясы
```

```
var k:integer; // берілген санға дейін өзгертін айнымалы
```

```
begin
```

```
    assert(a>0); //a берілген сан нөлден үлкен болса ғана есептейді
```

```
    if a=1 then
```

```
        begin
```

```
            result:=false;
```

```
            exit;
```

```
        end;
```

```
    result:=true;
```

```
    for k:=2 to a-1 do // k берілген санға дейінгі сандар
```

```
        begin
```

```
            if a mod k=0 then
```

```
                begin
```

```
                    result:=false;
```

```
                    exit;
```

```
                end;
```

```
        end;
```

```
end;
```

Жай сандардың санын есептейтін функцияның программасы:

```
function sany(a:integer):integer;
```

```
var k:integer;
```

```
begin
```

```
result:=0;
```

```
for k:=1 to a do // k берілген сан мен оған дейінгі барлық сандар
```

```
begin
```

```
    if JaibasarMysal(k) then // берілген санның жай сан екендігін
```

```
анықтау
```

```
        begin
```

```
            result:=Result+1; // берілген сан жай сан болса result өсіп отырады
```

```
        end;
```

```
    end;
```

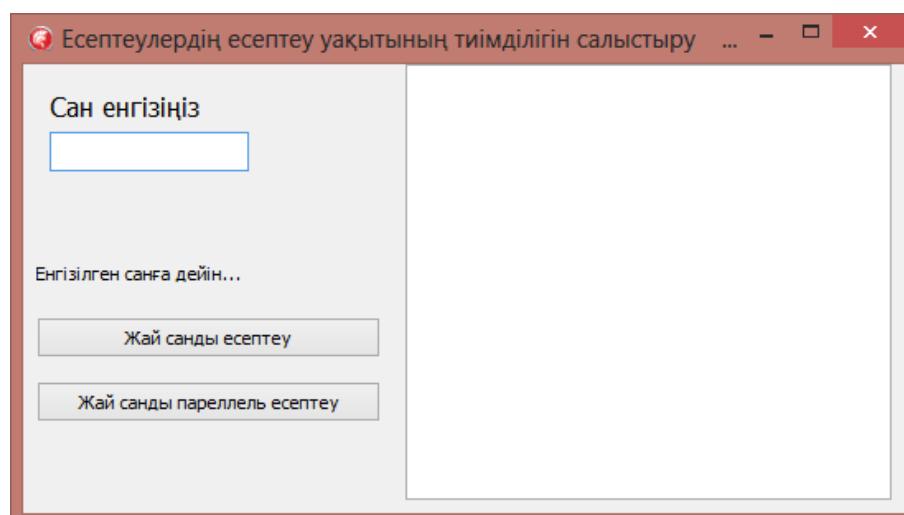
```
end;
```

Енді бұл функциялардың қызметін RadStudio ортасында есептеуді қарастыру. Алдымен *Task* қолданбай жай санды есептейтін, кейін *Task* қолданып жай санды параллель есептейтін программа құрастыру қажет.

Ол үшін төмендегі кестеден компоненттерді таңдап, олардың қасиеттерін өзгертіп, қандай оқиғаларына код жазатыныңызды қарауыңызға болады.

Компонент	Қасиеті	Қасиеттің мәні	Оқиғалар
Form1	Caption	Есептеулердің есептеу уақытының тиімділігін салыстыру	
Edit1	Text	Тазартыңыз	
Label1	Caption	Сан енгізіңіз	
Label2	Caption	Енгізілген санға дейін	
Memo1	Lines-TStrings	Тазартыңыз	
Button1	Caption	Жай санды есептеу	OnClick
Button2	Caption	Жай санды параллель есептеу	OnClick

Нәтижесінде форманы мына түрге келтіріңіз(сурет 43):



Сурет 43 - Form терезесіне компоненттерді орналастыру үлгісі

Енді батырмаларға код жазу қажет. Алдымен программа басында *System.Diagnostics*, *System.Threading* модульдарын қосып жазыңыз. Кейін жоғарыда жазған жай сандарды есептеудің тізбекті *Jaibasarmysal* және *Sany* функцияларын жазасыз. Әрі қарай *Button1*, *Button2* батырмаларына мынадай код жазып, компьютердің қанша уақытта есептейтіндігін көресіз[4], мұнда әр компьютердің есептеу жылдамдығы процессордың қуаты мен ядро санына байланысты өзгереді.

“Жай санды есептеу” *Button1* батырмасының коды:

```

procedure TForm1.Button1Click(Sender: TObject);
var
    Stopwatch:TStopWatch;    //жаңа уақытты есептеуге арналған
айнымалы
    Barlygu: Integer;        k:integer; //жай санды есептеуге арналған
айнымалы
    ElapsedSeconds: Double;  //қанша секунд есептейтіндігін білу

```



```

begin
k:=StrToInt(Edit1.Text);
Stopwatch:= TStopwatch.StartNew;
           // System.Diagnostics модулы құрамында
Barlygu:=Sany(k);
ElapsedSeconds := Stopwatch.Elapsed.TotalSeconds;
           //нәтижені Memo1 -ге шығару
Memo1.Lines.Add(Format('Берілген санға дейін %d жай сандар бар',
[Barlygu]));
Memo1.Lines.Add(Format('Бұл есептеу %:2f секундта есептелінді',
[ElapsedSeconds]));
end;

```

Жай санды *Task* қолданып параллель есептеуге арналған *Button2* батырмасының коды:

```

procedure TForm1.Button2Click(Sender: TObject);
var k:integer;
begin
k:=StrToInt(Edit1.Text);
TTask.Run(procedure           //параллель есептеу басы
var
Stopwatch: TStopWatch;           // System.Diagnostics құрамында
Barlygu: Integer;
ElapsedSeconds: Double;
begin
Stopwatch := TStopwatch.StartNew;
Barlygu:=Sany(k);
ElapsedSeconds := Stopwatch.Elapsed.TotalSeconds;
TThread.Synchronize(nil, procedure //синхрондау
begin
Memo1.Lines.Add(Format('Берілген санға дейін %d жай сандар бар',
[Barlygu]));
Memo1.Lines.Add(Format('Бұл есептеу %:2f секундта есептелінді',
[ElapsedSeconds]));
end);
end);           //параллель есептеу соңы
end;

```

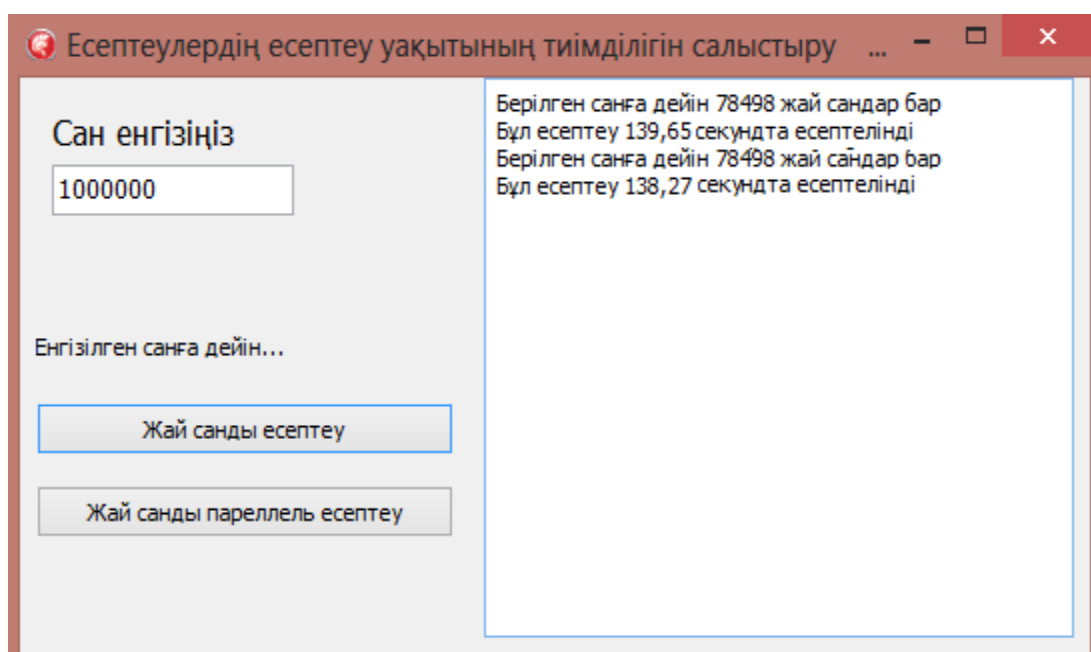
Бұл *Button2* кодында *PPL* синхронизациялану арқылы аутоматты түрде компьютердің барлық мүмкіндігін қолданады. Есептеу уақытын бақылау мақсатында осы қосымшаның сипаттамасына әр түрлі бірнеше инфрақұрылымдарда есептеу арқылы эксперимент жүргізуге болады. Мысалы, біздің эксперимент екі түрлі инфрақұрылымдарда жүргізілді.

Бірінші эксперимент бір ядролы процессорда жүргізілді.

Кесте10 - Бірінші инфрақұрылым

Ноутбук	
Процессор	Intel Core i5-3210M CPU 2.50GHz
Жады	6 GB
Операциялық жүйе	Microsoft Windows 8
Құрастыру ортасы	RadStudio XE7

Нәтижесі (сурет 44):



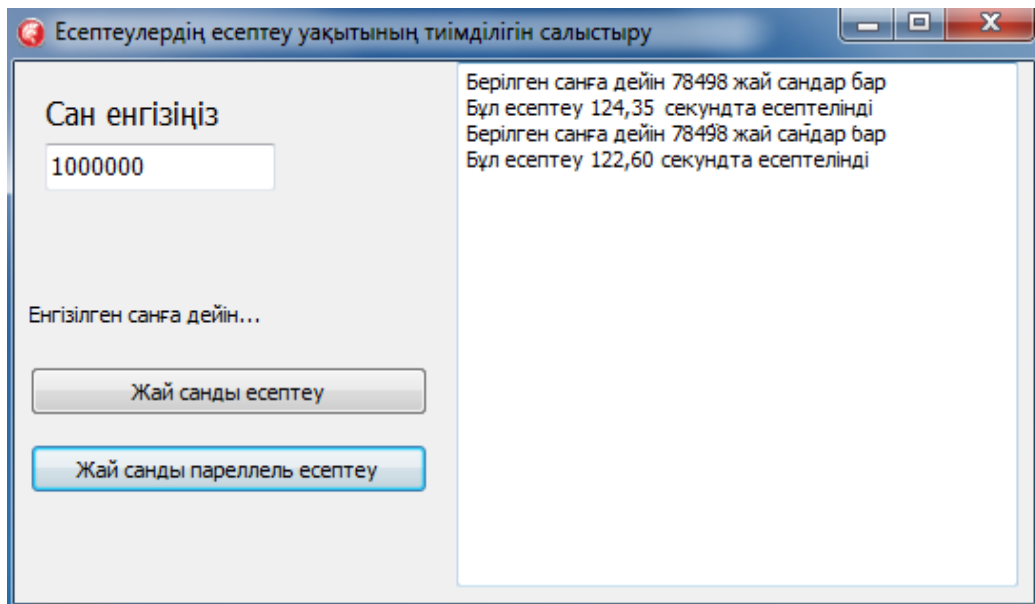
Сурет 44 - Бірінші инфрақұрылымда есептеу нәтижелері

Бұл есептеулер бір ядролы процессорда бірінші рет дәстүрлі тізбекті түрде, ал екінші рет *Task* қолданылып есеп параллель есептелді.

Ал келесі эксперимент екінші инфрақұрылым екі ядролы процессорда жүргізілді (сурет 45).

Кесте 11 - Екінші инфрақұрылым

Компьютер	
Процессор	Pentium Dual-Core CPU 3.07GHz
Жады	4 GB
Операциялық жүйе	Microsoft Windows 7
Құрастыру ортасы	RadStudio XE7



Сурет 45 - Екінші инфрақұрылымда есептеу нәтижелері

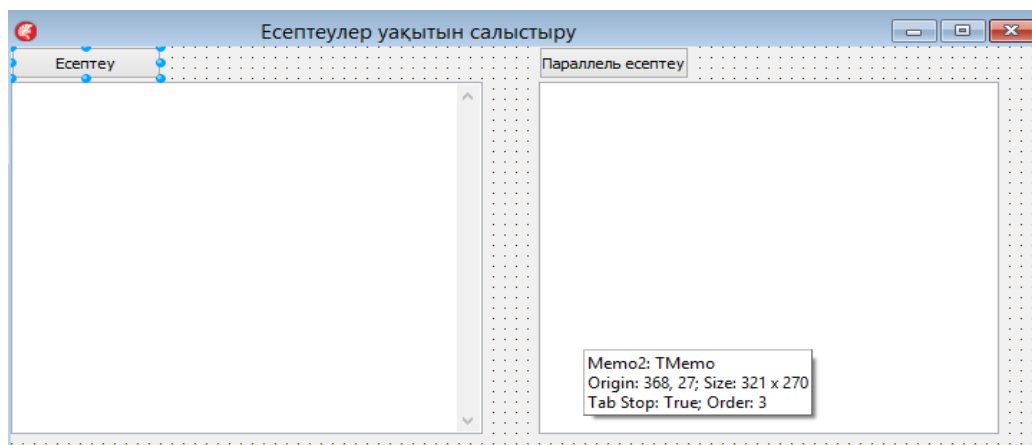
Сіздер өз компьютерлеріңізде осы есептердің шешу уақытымен осы нәтижелердің есептеу уақытын салыстырыңыз.

Есеп 2

Үлкен санмен жұмыс істейтін, көп итерациядан тұратын мысалды қарастырайық. Ол үшін Академик Асқар Жұмаділдаевтің «Үшінші шешімі бар, балалар соны тапса деймін. Төртінші шешімі бар ма, жоқ па?...»[37], - деп балалардың математикаға қызығушылығын арттыруға бағытталған берілген $n!+1=m^2$ Броккард есебіне программа құрастырайық. Есептің шешімдерін анықтау мақсатында Object Pascal тілінің ұзын бүтін санының типі cardinal қолданайық. Cardinal типі 4294967295 санға дейін есептей алады. Әрі қарай факториал есептеу үшін *fact* функциясын жазыңыз:

```
FUNCTION FACT(K:integer):cardinal;
BEGIN
  IF K=1 THEN FACT:=1 ELSE FACT:=K*FACT(K-1);
END;
```

Енді бұл функциялардың қызметін *RadStudio* ортасында есептеуге болады. Алдымен тізбектей есептеп, кейін *Task* қолданып, параллель есептейтін программа құрастырып, есептеу уақыттарын салыстырайық. Ол үшін жаңа жоба құрастырыңыз да, форманы келесі түрге келтіріңіз (сурет 46):



Сурет 46 - Form терезесіне компоненттерді орналастыру үлгісі

Компоненттер мен олардың қасиеттерін өзгертіп, қандай оқиғаларына код жазатыныңызды төмендегі кестеден қараңыз:

Компонент	Қасиеті	Қасиеттің мәні	Оқиғалар
Form1	Caption	Есептеулер уақытын салыстыру	
Memo1	Lines-TStrings	Тазартыңыз	
Memo2	Lines-TStrings	Тазартыңыз	
Button1	Caption	Есептеу	OnClick
Button2	Caption	Параллель есептеу	OnClick

Программа басында *System.Diagnostics*, *System.Threading* модульдарын қосып жазуды ұмытпаңыз. Әрі қарай батырмалардың кодын жазыңыз.

«Есептеу» Button1 батырмасының коды:

```

procedure TForm1.Button1Click(Sender: TObject);
var
  Stopwatch:TStopWatch;
  y,j,P:cardinal; n,i:integer;;
  ElapsedSeconds: Double;
Begin
  Stopwatch := TStopwatch.StartNew;
  n :=12; // cardinal типі 12! ға дейін дұрыс есептей алады
  FOR I:= 1 TO N DO
  for y:=1 to 65536 do
    //65536 квадраты 4294967295 болғандықтан одан үлкен санды есептей
    алмаймыз
    BEGIN
      P:=FACT(I);
      j:=y*y-1;
      if p=j then begin
        ElapsedSeconds := StopWatch.Elapsed.TotalSeconds;
        //нәтижені Memo1 -ге шығару

```

```

Memo1.Lines.Add(Format('%d!=',[i])+(Format('%d',[p])));
Memo1.Lines.Add(Format('%d^2-1=',[y])+(Format('%d',[j])));
Memo1.Lines.Add(Format('Бұл есептеу %:2f секундта есептелінді',
[ElapsedSeconds]));
end;
end;
END;

```

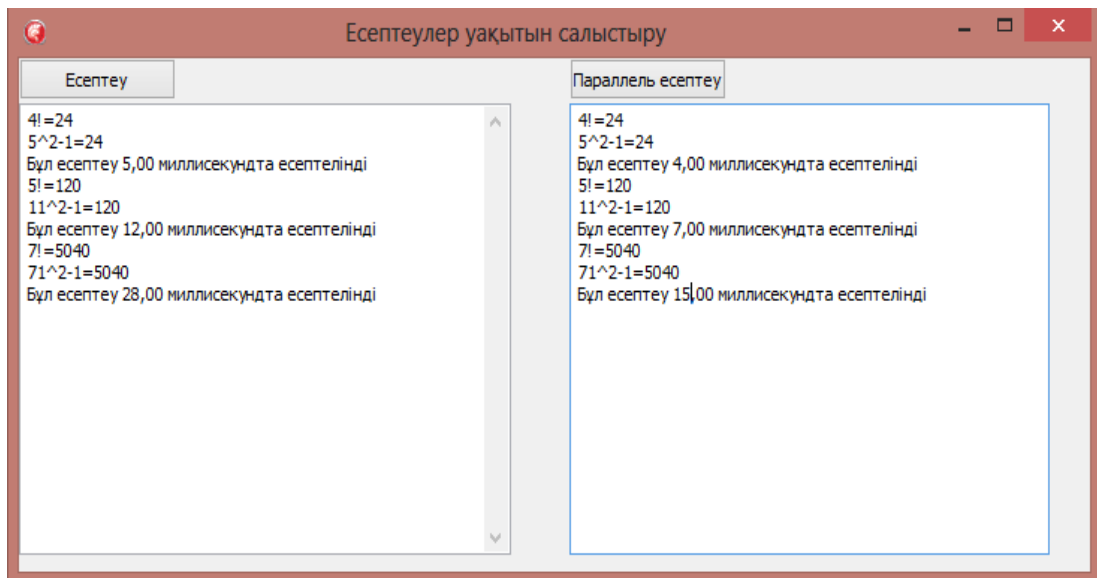
«Параллель есептеу» Button2 батырмасының коды:

```

procedure TForm1.Button2Click(Sender: TObject);
BEGIN
    TTask.Run(procedure // параллель есептеулер
        Var Stopwatch: TStopWatch;
            y,j,P:cardinal; n, i:integer;
            ElapsedSeconds: Double;
        begin
            Stopwatch := TStopwatch.StartNew;
            n :=12;
            FOR I:= 1 TO N DO
                for y:=1 to 65536 do
                    begin
                        P:=FACT(I);
                        j:=y*y-1;
                        if p=j then begin
                            ElapsedSeconds := Stopwatch.Elapsed.TotalSeconds;
                            TThread.Synchronize
                                (nil, procedure //ақпаратты синхрондау
                                    //нәтижені Мемо2 -ге шығару
                                    begin
                                        Memo2.Lines.Add(Format('%d!=',[i])+(Format('%d',[p])));
                                        Memo2.Lines.Add(Format('%d^2-1=',[y])+(Format('%d',
[j])));
                                        Memo2.Lines.Add(Format('Бұл есептеу %:2f секундта
есептелінді', [ElapsedSeconds]));
                                    end);
                                end;
                            end;
                        end;
                    end);
                end);
            END;

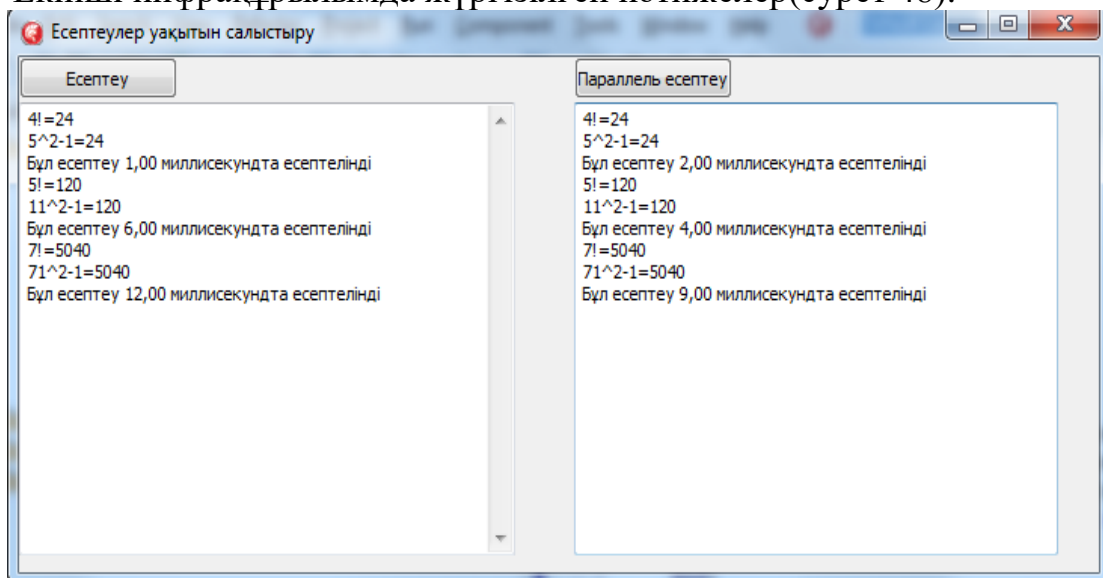
```

Бірінші инфрақұрылымда орындалған нәтижелер(сурет 47):



Сурет 47 - Бірінші инфрақұрылымда орындалған нәтижелер

Екінші инфрақұрылымда жүргізілген нәтижелер(сурет 48):



Сурет 48 - Екінші инфрақұрылымда орындалған нәтижелер

Есептеу нәтижесінен 4294967295 санға дейін тек үш шешімі ғана бар екенін дәлелдедік(4,5), (5,11),(7,71).

Осы эксперименттерден RadStudio XE7 ортасында жалпы есептеулерден қарағанда *Task* класын қолданып құрастырған параллель есептеулердің орындалу уақыты аз екенін көруге болады.

Өздік жұмыс:

Күрделі итерациядан тұратын есептер таңдап, оған параллель есептеудің *TTask* класын қолданатын программа құрастырып, нәтижесін салыстырыңыз.

Бақылау сұрақтары:

1. *TTask* класы қандай модуль құрамында?

2. Тізбектей есептеумен параллель есептеудің есептеу уақыттары қалай өзгереді?

Әдістемелік нұсқау

4.1- бөліміндегі теориялық материалдармен танысып, берілген жұмысты орындау

Қолданылған әдебиеттер

Nick Hodges. Coding in RadStudio , Nepeta Enterprises 2014 - 2015

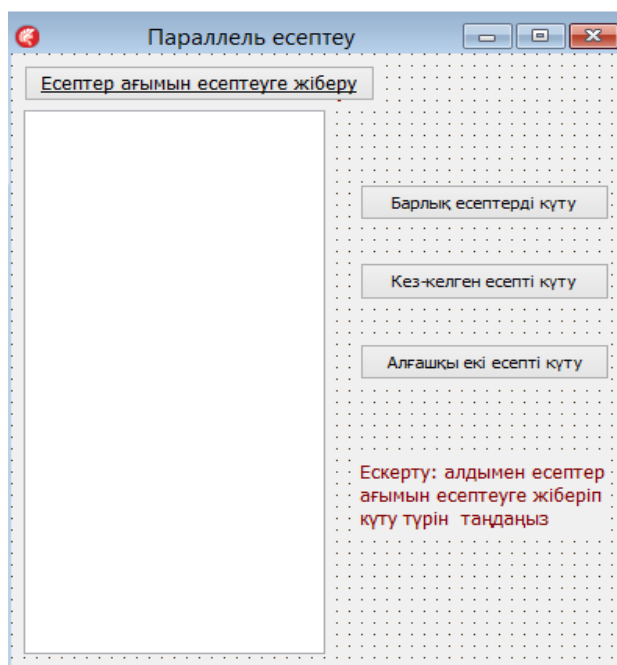
Тақырып 8. RadStudio ортасында параллель есептеулер.

Мақсаты: RadStudio XE7 ортасында кодтың параллель орындалуында күтуді ұйымдастыру үшін қолданылатын қызметтермен таныстыру және оны жүзеге асыру.

Есептің қойылуы: күту қызметін TTask класы арқылы ұйымдастыру.

Жұмыс реті: Бір уақытта бірнеше есептерді есептеу қажет болған кезде бірнеше ядролы процессордың қызметін тиімді қолдану немесе кейбір есептеулер ағымының нәтижесі шыққанша күту немесе көре беру мүмкіндігін TTask класы арқылы жүзеге асыруға болады. Біз ол қызметтерді көрсету үшін алдыңғы тәжірбиелік жұмыстағы Jaibasarmysal және Sany функцияларын қолданайық.

Жаңа жоба құрастыру үшін форманы мына түрге келтіріңіз (сурет 49).



Сурет 49 - Form терезесіне компоненттерді орналастыру үлгісі

Ол үшін компоненттер мен олардың қасиеттерін өзгертіп, қандай оқиғаларына код жазатыныңызды төмендегі кестеден қараңыз.

Компонент	Қасиеті	Қасиеттің мәні	Оқиғалар
Form1	Caption	Параллель есептеу	
Label1	Caption	Ескерту: алдымен есептер ағымын есептеуге жіберіп күту түрін таңдаңыз	
	Font	Color->Marvin, Шрифт->20	
Memo1	Lines-TStrings	Газартыңыз	
Button1	Caption	Есептер ағымын есептеуге жіберу	OnClick
Button2	Caption	Барлық есептерді күту	OnClick
Button3	Caption	Кез-келген есепті күту	OnClick
Button4	Caption	Алғашқы екі есепті күту	OnClick

Программа басында *System.Diagnostics*, *System.Threading* модульдарын қосып жазыңыз. Алдымен *Unit1* модулына жоғарыда жазған жай сандарды есептеудің дәстүрлі функцияларын жазыңыз *function Jaibasarmysal* және *function Sany*.

Әрі қарай *Button1*, *Button2*, *Button3*, *Button4* батырмаларына келесі кодтарды жазып, өз компьютеріңіздің қанша уақытта есептейтіндігін көріңіз.

«Есептер ағымын есептеуге жіберу» *Button1* батырмасының коды:

```

procedure TForm1.Button1Click(Sender: TObject);
var
  Stopwatch:TStopWatch;
  ElapsedSeconds: Double; Barlygy:integer;
begin
  AllTasks[0] := TTask.Run(procedure // бірінші ағымды жіберу
  begin
    Stopwatch := TStopwatch.StartNew;
    Barlygy:=sany(300000);
    //300000 дейінгі сандардың ішінен жай сандарды анықтау
    ElapsedSeconds := StopWatch.Elapsed.totalSeconds;
    TThread.Synchronize(TThread.CurrentThread,
    procedure
    begin
      Memo1.Lines.Add('300000 ға дейін есептеу аяқталды');
      Memo1.Lines.Add(Format('Бұл %:2f секунд есептеді',
      [ElapsedSeconds]));
    end);
  end);
  AllTasks[1] := TTask.Run(procedure // екінші ағымды жіберу
  begin
    Barlygy:=Sany(400000);
  
```



```

//400000 дейінгі сандардың ішінен жай сандарды анықтау
ElapsedSeconds := Stopwatch.Elapsed.totalSeconds;
    TThread.Synchronize(TThread.CurrentThread,
    procedure
    begin
    Memo1.Lines.Add('400000 ға дейін есептеу аяқталды');
    Memo1.Lines.Add(Format('Бұл %:2f секунд есептеді',
[ElapsedSeconds]));
    end);
    end);
AllTasks[2] := TTask.Run(procedure // үшінші ағымды жіберу
begin
Barlygy:=Sany(500000);
//500000 дейінгі сандардың ішінен жай сандарды анықтау
ElapsedSeconds := Stopwatch.Elapsed.totalSeconds;
    TThread.Synchronize(TThread.CurrentThread,
    procedure
    begin
    Memo1.Lines.Add('500000 ға дейін есептеу аяқталды');
    Memo1.Lines.Add(Format('Бұл %:2f секунд есептеді',
[ElapsedSeconds]));
    end);
    end);
end;

```

«Барлық есептерді күту» Button2 батырмасының коды:

```

procedure TForm1.Button2Click(Sender: TObject);
begin
    TTask.Run(procedure
    begin
    TTask.WaitForAll(AllTasks);
    TThread.Synchronize(nil,
    procedure
    begin
    Memo1.Lines.Add('БАРЛЫҒЫ ЕСЕПТЕЛДІ');
    end);
    end);
end;

```

«Кез-келген есепті күту» Button3 батырмасының коды:

```

procedure TForm1.Button3Click(Sender: TObject);
begin
    TTask.Run(procedure
    begin

```

```

        TTask.WaitForAny(AllTasks);
        TThread.Synchronize(nil,
        procedure
        begin
        Memo1.Lines.Add('КЕЗ-КЕЛГЕНІН КҮТІП СОҒЫ
АЯҚТАЛДЫ');
        end);
        end);
    end;

```

«Алғашқы екі есепті күту» Button4 батырмасының коды:

```

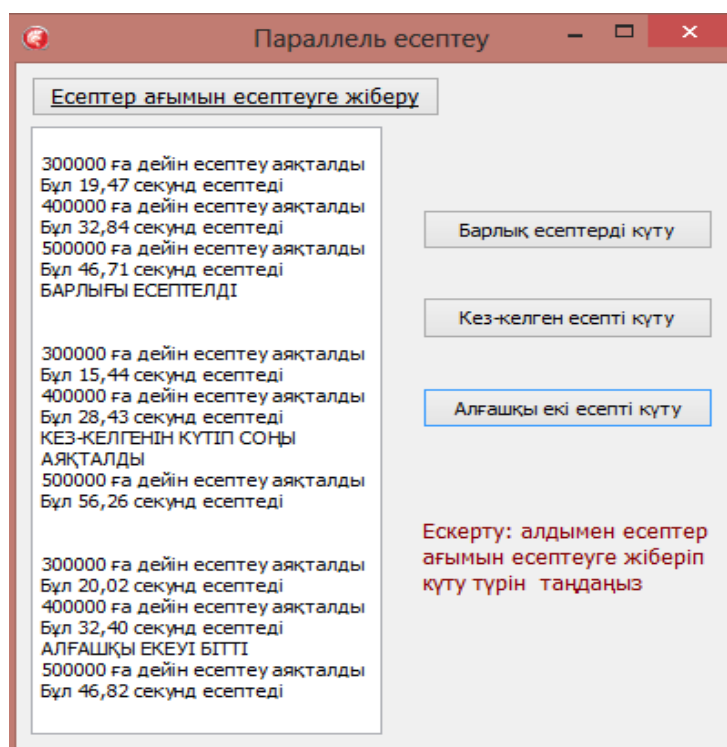
procedure TForm1.Button4Click(Sender: TObject);
begin
    TTask.Run(procedure
    var
        SomeTasks: array[0..1] of ITask; // есептер массивы
    begin
        SomeTasks[0] := AllTasks[0];
        SomeTasks[1] := AllTasks[1];
        TTask.WaitForAll(SomeTasks);
        TThread.Synchronize(nil,
        procedure
        begin
        Memo1.Lines.Add('АЛҒАШҚЫ ЕКЕУІ БІТТІ');
        end);
        end);
    end;
end;

```

Кесте 12 - Тестлеу инфрақұрылымы

Компьютер	
Процессор	Intel Core i5-3210M CPU 2.50GHz
Жады	6 GB
Операциялық жүйе	Microsoft Windows 8
Құрастыру ортасы	RadStudio XE7

Нәтижесі:



Сурет 50 - Form терезесіне компоненттерді орналастыру үлгісі

Нәтижеден есептеу уақыты күтудің қажеттілігіне байланысты әр түрлі өзгеретінін байқауыңызға болады.

Өздік жұмыс:

Күрделі итерациядан тұратын есептер таңдап, оған кодтың параллель орындалуы TTask классының күту қызметі арқылы жүзеге асатын программа құрастырып нәтижесін салыстырыңыз.

Бақылау сұрақтары:

1. Бұл мысалда алдымен не жіберіліп, кейін күту ұйымдастырылған. Ал алдымен *Task* арқылы күту ұйымдастырылып, кейін кодтың параллель орындалуын жіберуге болатындай RadStudio ортасында программа құруға бола ма?

2. Егер болатын болса түсіндіріңіз?

Әдістемелік нұсқау

Теориялық материалдармен танысып, берілген жұмыстарды орындау.

Қолданылған әдебиеттер

Nick Hodges. Coding in RadStudio , Nepeta Enterprises 2014 - 2015

Тақырып 9. RadStudio ортасында кроссплатформалық қосымшалар құруда параллель есептеулер қолдану.

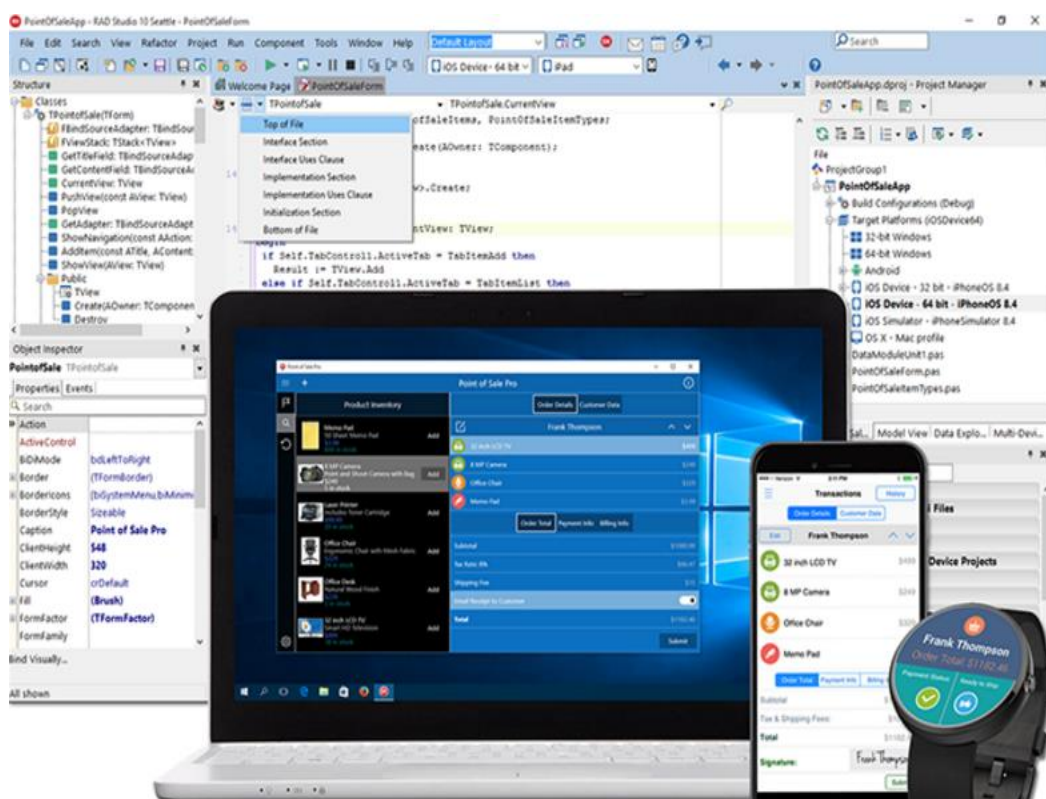
Мақсаты: RadStudio XE7 ортасында кроссплатформалы қосымша құру мен параллель есептеулерді қолдануды

Есептің қойылуы: RadStudio XE7 кроссплатформалы қосымша құрыңыз

Жұмыс реті:

Мысал 1.

RadStudio XE7 ортасында құрастырылған қосымшаны қолдану кезінде құрылғылар әр түрлі болса да (ноутбук, телефон, планшет, компьютер), олардың процессорларының ядроларының бар мүмкіндігін қолданатындай программа құруға болады, басқаша айтқанда жоғары өнімді VCL және *FireMonkey* қосымшаларын құруға болатындығын атап өттік(сурет 59).



Сурет 59 - RadStudio XE7 ортасында кроссплатформалы қосымшалар құру

Енді осыған мысал ретінде планшет немесе ұялы телефонға арналған қосымшаға параллель есептеу қолданатын программа құрастыру қажет. Ол үшін құрылғыда орнатылған Android операциялық жүйесі мен RadStudio ортасын баптап оған параллель есептеу қолданылатын мысал келтірейік.

1. Android-нен RAD Studio XE7 (RadStudio , C++ Builder) ортасын баптау

Алдымен RAD Studio XE7 ортасын байланыстыратын Android орнатылған планшет немесе ұялы телефон, USB-сымдарын дайындаймыз. Онымен қоса компьютеріңізде RAD Studio XE7 орнатылған кезде Android

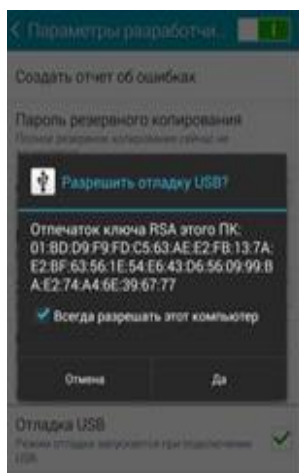
SDK және Android NDK кітапханаларының орнату белгісіне “Ия” деп көрсету қажет. Егер Android SDK және Android NDK кітапханалары орнатылмаған болса, компьютердің басқару панеліне кіріп, баптаудан бұл екі кітапханаларды орнату керек.

2. Компьютерге құрылғының драйверін орнату

Компьютер мен планшетті сыммен қосып, планшеттің «Баптау» (Настройка) бөлімінен «Құрылғы туралы» (Об устройстве) бөлімінің «Құрастыру нөмірін» (номер сборки) тауып, оған 7 рет шертеміз. Осыдан кейін «Құрастырушы параметрлері» (Параметры разработчика) жасырын бөлімі пайда болады. Осыдан «Отладка USB» бөліміне кіріп белгі қоямыз. Бұл белгі қойылғаннан кейін компьютерге сол құрылғының драйвері аутоматты түрде орнатылады. Егер аутоматты түрде орнатылмаса, онда Интернеттен сол құрылғының драйверін жүктеп орнату керек болады. Құрылғы орнатылғаннан соң USB сымын үзіп, қайтадан компьютерге қайтадан қосу керек.

3. USB отладкасын рұқсатын ету

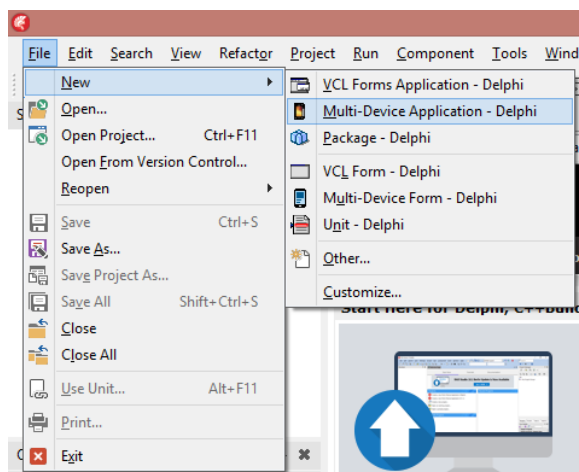
Осыдан кейін планшетте(сурет 60) «USB отладка рұқсатын ету» (Разрешить отладку USB) хабарламасы аутоматты түрде шығады. Оған «әрқашан осы компьютерге рұқсат» (Всегда разрешать этот компьютер) белгісін көрсетіп “Ия” (Да) батырмасын басамыз.



Сурет 60 - Планшетті баптау

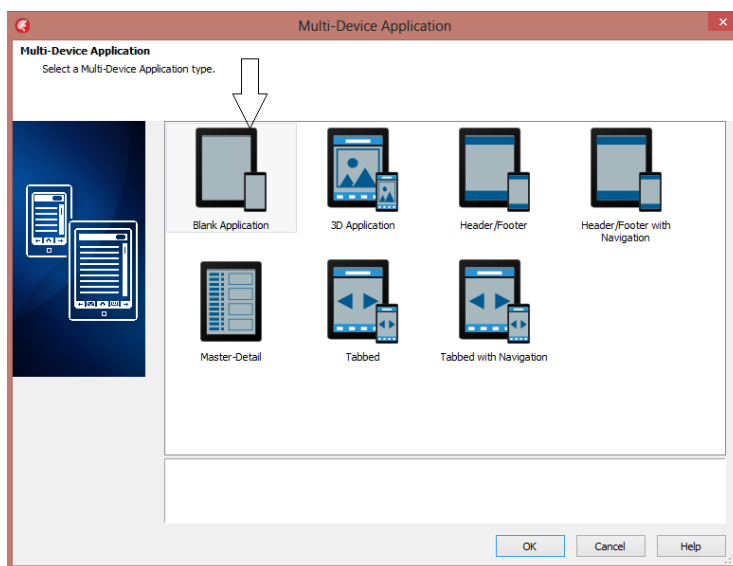
4. RAD Studio ортасын жүктеу

Осы баптаулар орнатылғаннан кейін ғана *RAD Studio ортасын* жүктейміз де, кроссплатформалы қосымша құру үшін *File->New->Multi-Device application -RadStudio* (сурет 61) командаларын орындайсыз.



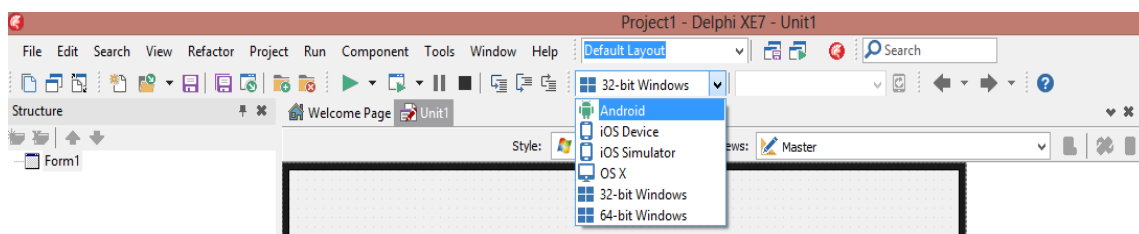
Сурет 61 - Кроссплатформалы қосымша құру

Шыққан терезеден *Blank Application* таңдап ОК басасыз (сурет 62).



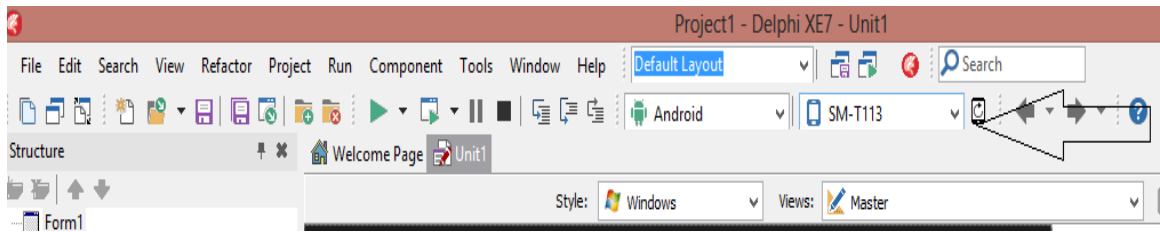
Сурет 62 - Blank Application

Әрі қарай құралдар тақтасынан қажетті операциялық жүйе түрін таңдайсыз. Біздің жағдайда Android (сурет 63).



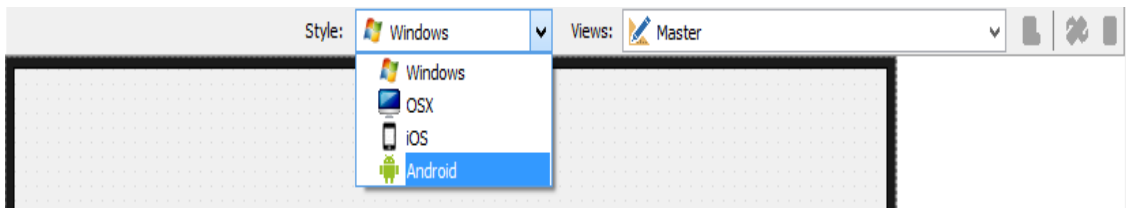
Сурет 63 - Қажетті операциялық жүйенің түрін таңдау

Егер баптаулар дұрыс болса және планшетті немесе ұялы телефонды дұрыс қоссаңыз, онда жанындағы таңдаудан сол құрылғының атауы шығады. Егер шықпаса, ұялы телефон суреті тұрған *Refresh Device* командасын таңдап көріңіздер. Біздің жағдайда SM-T113 құрылғысы шықты және қосымша құруға дайын (сурет 64).



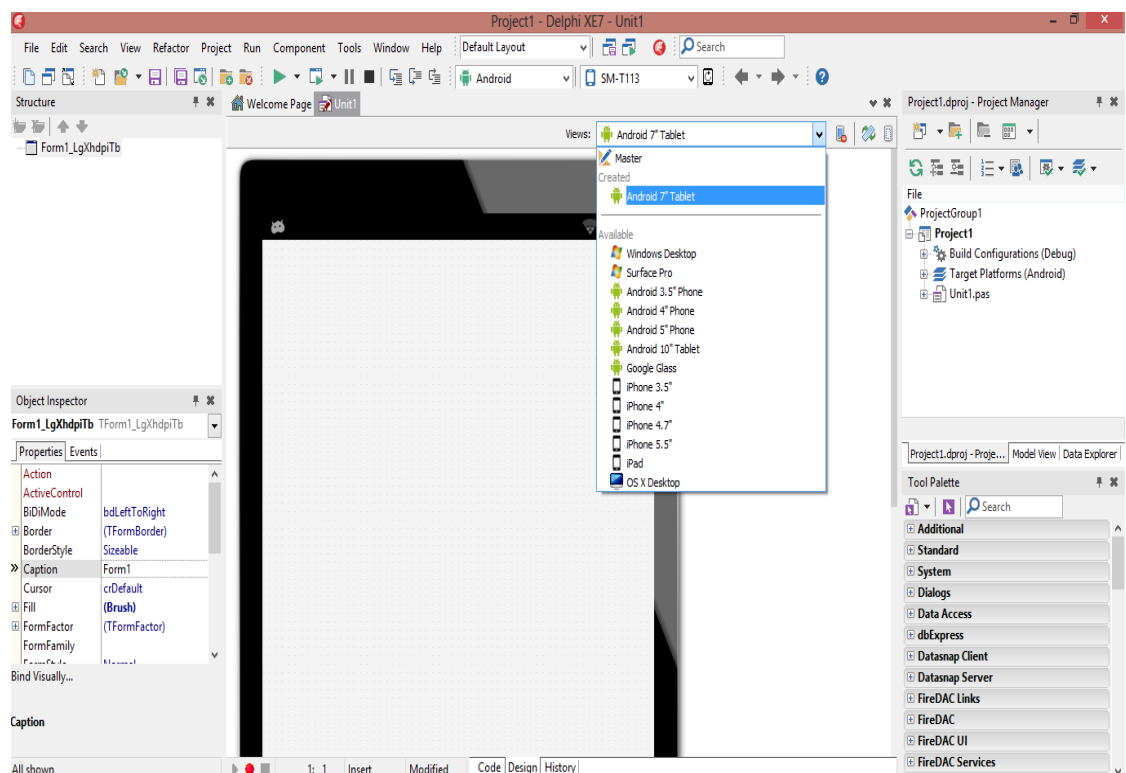
Сурет 64 - Қажетті құрылғының моделін таңдау

Әрі қарай Style бөлімінен де, Android таңдайсыз (сурет 65).



Сурет 65 - Операциялық жүйені таңдау

Ал құрылғыңызды көрсету үшін Views бөлімінен өз құрылғыңыздың түрін таңдайсыз (сурет 66).

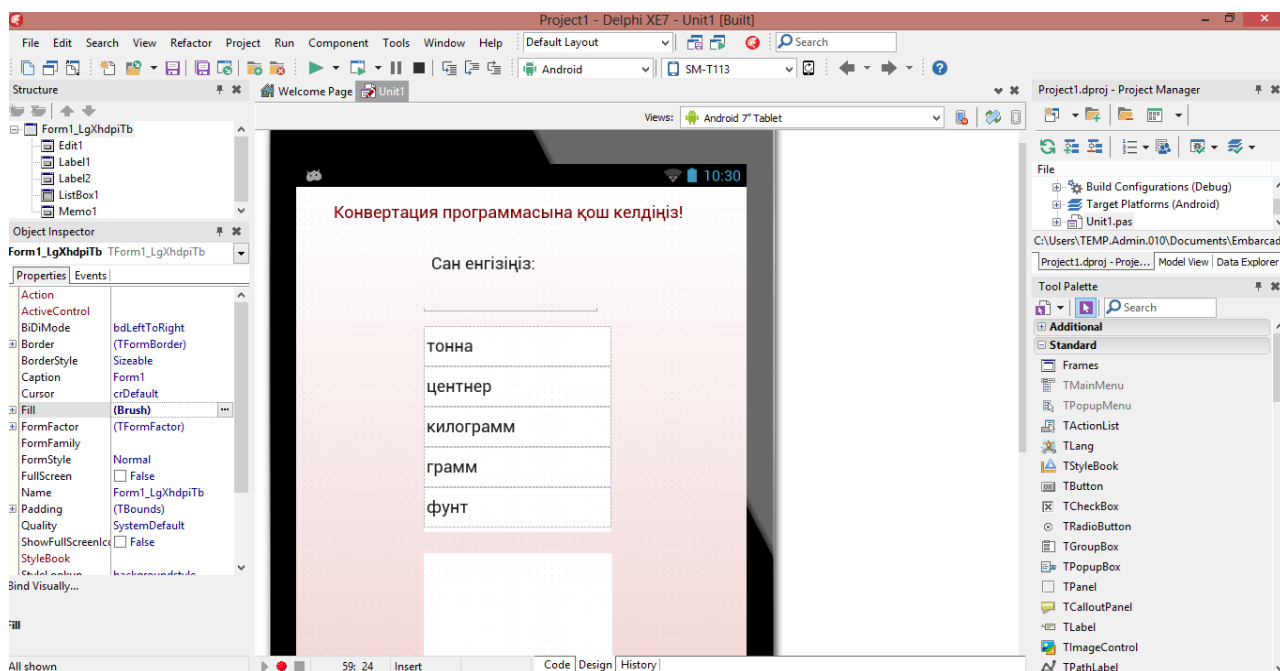


Сурет 66 - Қажетті құрылғының пішінін таңдау

Осылайша Radstudio ортасы Android - қа қосымша құруға дайын болды.

5. Android құрылғысына арналған конвертация программасын құрастыру

Кроссплатформалы қосымша құруда параллель есептеуді көрсету үшін конвертация программасын (сурет 67) құрастырайық.



Сурет 66 -Конвертация программасын жобалау түрі

Ол үшін құрастыру терезесін төмендегі кестеге қарап компоненттерді таңдап қасиеттерін өзгертіңіз:

Компонент	Қасиеті	Қасиеттің мәні	Оқиғалар
Form1_LgXhdpiTb	Caption	Есептеулердің есептеу уақытының тиімділігін салыстыру	
Edit1	Text	Газартыңыз	
Label1	Text	Конвертация программасына қош келдіңіз!	
	Font->Color	Maroon	
Label2	Text	Сан енгізіңіз	
Memo1	Lines-TStrings	Газартыңыз	
ListBox1	Lines-TStrings	тонна центнер килограмм грамм фунт	ListBox1ItemClick
Button1	Text	Шығу	OnClick

Әрі қарай *ListBox1* компонентінің *ListBox1ItemClick* оқиғасына мынадай код жазу керек:

```
procedure TForm1.ListBox1ItemClick(const Sender: TCustomListBox;
const Item: TListBoxItem);
```



```

begin
  TTask.Run(procedure //параллель есептеулер
    var x,y1,y2,y3,y4,y5:real;
    begin
      x:=strToFloat(Edit1.Text); //енгізу
      begin

        //өлшем бірлікті таңдау
        Case ListBox1.ItemIndex of

          0: begin y1:=x; y2:=x*10; y3:=x*1000; y4:=x*1000000; y5:=x*2204.6; end; // тонна

          1: begin y1:=x*0.1; y2:=x; y3:=x*100; y4:=x*100000; y5:=x*244.1928; end; //центнер

          2: begin y1:=x*0.001; y2:=x*0.1; y3:=x; y4:=x*1000; y5:=x*2.441928; end; //кг

          3: begin y1:=x*0.000001; y2:=x*0.00001; y3:=x*0.001; y4:=x; y5:=x*0.00244193; end;
//гм

          4: begin y1:=x*0.00040951; y2:=x*0.00409512; y3:=x*0.40951241; y4:=x*409.5124;
y5:=x; end; //фунт

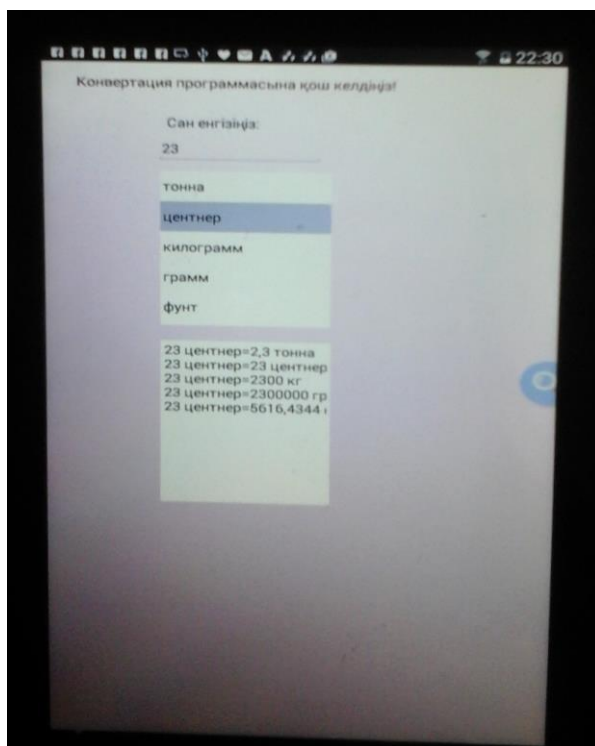
        end;

        //нәтижесін шығару
        Memo1.Lines.Add(FormatFloat(#####.#####',x)+ListBox1.Items[Listbox1.ItemIndex]+
FormatFloat(=#####.##### тонна',y1));
        Memo1.Lines.Add(FormatFloat(#####.#####',x)+ListBox1.Items[Listbox1.ItemIndex]+
FormatFloat(=#####.##### центнер',y2));
        Memo1.Lines.Add(FormatFloat(#####.#####',x)+ListBox1.Items[Listbox1.ItemIndex]+
FormatFloat(=#####.##### кг',y3));
        Memo1.Lines.Add(FormatFloat(#####.#####',x)+ListBox1.Items[Listbox1.ItemIndex]+
FormatFloat(=#####.##### грамм',y4));
        Memo1.Lines.Add(FormatFloat(#####.#####',x)+ListBox1.Items[Listbox1.ItemIndex]+
FormatFloat(=#####.##### фунт',y5));
        end;
        end); // параллель есептеулер соңы
        end;
end.

```

Ескерту: Программаны Android-қа жіберуден бұрын Windows-та орындап, қателерімен жұмысты жасап алған дұрыс. Себебі, Android-қа жіберіп қателерін дұрыстау көп уақыт алады.

Планшетте параллель есептелген конвертация программа нәтижесін 67-суреттен көре аласыз.



Сурет 67 - Планшетте параллель есептелген конвертация программа нәтижесі

Өздік жұмыс:

Бұл программаға қосымша ұзындықты, уақытты, ауданды және көлемді параллель есептейтін программа жазыңыз.

Әдістемелік нұсқау

Теориялық материалдармен танысып, берілген жұмыстарды орындау.

Қолданылған әдебиеттер

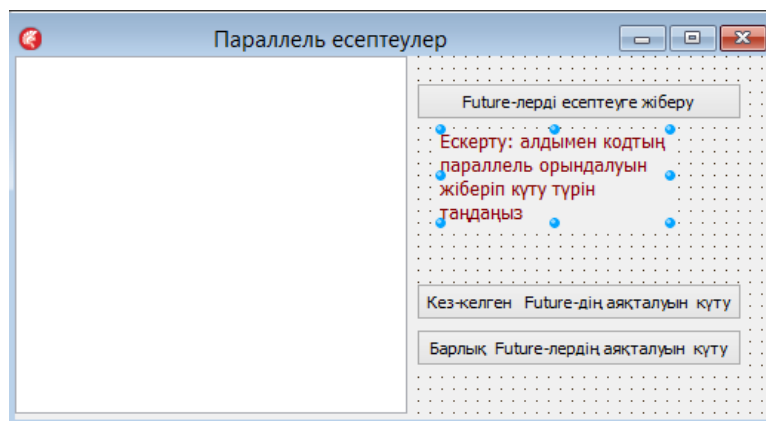
1. <https://habrahabr.ru/company/RadStudio/blog/253929/>
2. [docwiki.embarcadero.com/RADStudio/XE7 // 21/12/2016](http://docwiki.embarcadero.com/RADStudio/XE7//21/12/2016)

Тақырып 10. Параллельдеудегі проблемалар

Мақсаты: *RadStudio XE7 ортасында кодтың параллель орындалуында күту қызметін Future арқылы жүзеге асыруды үйрету.*

Есептің қойылуы: күту қызметін Future класы арқылы ұйымдастыру.

Жұмыс реті: Көп есептеулердің барлық нәтижесі шыққанша күту мүмкіндігін Future класы арқылы да жүзеге асыруға болады. Біз ол қызметтерді көрсету үшін алдыңғы мысалдардағы JaibasarMysal және Sany функцияларын қолданайық. Жаңа жоба құрастыру үшін форманы мына түрге келтіріңіз (сурет 51):



Сурет 51 - Form терезесіне компоненттерді орналастыру үлгісі

Ол үшін компоненттер мен олардың қасиеттерін өзгертіп, қандай оқиғаларына код жазатыныңызды төмендегі кестеден қараңыз:

Компонент	Қасиеті	Қасиеттің мәні	Оқиғалар
Form1	Caption	Параллель есептеулер	
Label1	Caption	Ескерту: алдымен Future-лерді есептеуге жіберіп күту түрін таңдаңыз	
	Font	Color->Marvin, Шрифт->20	
Memo1	Lines-TStrings	Тазартыңыз	
Button1	Caption	Future-лерді есептеуге жіберу	OnClick
Button2	Caption	Кез-келген Future-дің аяқталуын күту	OnClick
Button3	Caption	Барлық Future-лердің аяқталуын күту	OnClick

Программа басында *System.Threading* модульдарын қосып жазыңыз. Алдымен *Unit1* модулына жоғарыда жазған жай сандарды есептеудің дәстүрлі функцияларын жазамыз: *function Jaibasarmysal* және *function Sany*.

Әрі қарай *Button1*, *Button2*, *Button3* батырмаларына мынадай код жазамыз.

«Future-лерді есептеуге жіберу» *Button1* батырмасының коды:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    Result400000 := TTask.Future<integer>(function: integer
begin
    Result := Sany(400000);
end);
Memo1.Lines.Add('400000 орындалу басталды');
Futures[0] := Result400000;
```

```

Result450000 := TTask.Future<integer>(function: integer
begin
Result := Sany(450000);

                                end);
Memo1.Lines.Add('450000 орындалу басталды');
Futures[1] := Result450000;
Result500000 := TTask.Future<integer>(function: integer
begin
Result := Sany(500000);

                                end);
Memo1.Lines.Add('500000 орындалу басталды');
Futures[2] := Result500000;
end;

```

«Кез-келген Future-дің аяқталуын күту» Button2 батырмасының коды:

```

{ мұндағы кодтан WaitFor әдісі қолданылмай, тек келесі есептеудің
есептелуін күтуді GetValue блока қойып тұрғанын байқауға болады. }

procedure TForm1.Button2Click(Sender: TObject);
begin
Memo1.Lines.Add('400,000 дейін' + Result400000.GetValue.ToString
+ 'жай сандар бар');
Memo1.Lines.Add('Күту аяқталды. Бірінші нәтиже есептелді');
Memo1.Lines.Add('450,000 дейін' + Result450000.GetValue.ToString
+'жай сандар бар' );
Memo1.Lines.Add('500,000 дейін' + Result500000.GetValue.ToString
+'жай сандар бар' );
end;

```

«Барлық Future-лердің аяқталуын күту» Button3 батырмасының коды:

```

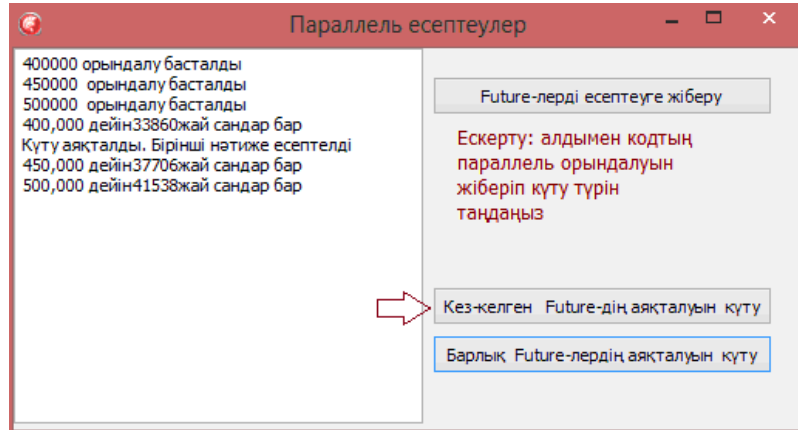
{ мұндағы кодтан WaitForAll әдісі қолданылып, келесі есептеулерді
күтуді GetValue блока қойып тұрып 3 нәтиже бір уақытта шығатынын
байқауға болады. }

procedure TForm1.Button3Click(Sender: TObject);
begin
TFuture<integer>.WaitForAll(Futures);
Memo1.Lines.Add('Күту аяқталды. Бұл хабарлама барлық нәтижемен
бірге шығу керек');
Memo1.Lines.Add('400,000 дейін' + Result400000.GetValue.ToString +
'жай сандар бар');
Memo1.Lines.Add('450,000 дейін' + Result450000.GetValue.ToString
+'жай сандар бар' );

```

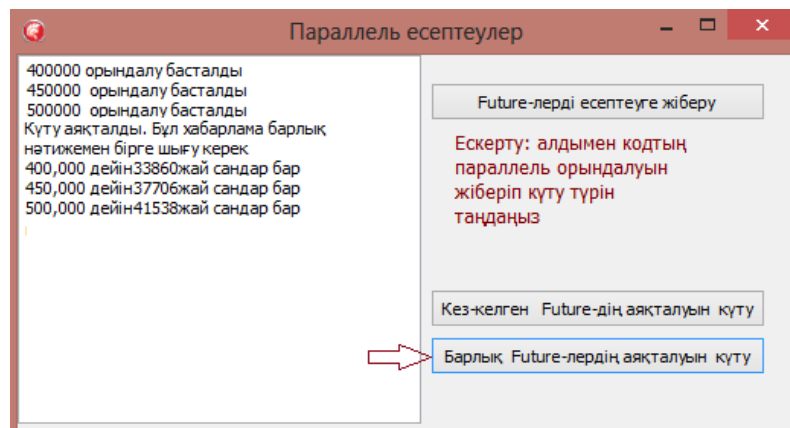
```
Memo1.Lines.Add('500,000 дейін' + Result500000.GetValue.ToString  
+'жай сандар бар');  
end;
```

Кез-келген есептеулердің аяқталуын күту нәтижелері (сурет 52):



Сурет 52 - Кез-келген есептеулердің аяқталуын күту нәтижелері

Барлық есептеулерді күтіп нәтижесін бір уақытта шығару нәтижелері (сурет 53):



Сурет 53 - Барлық есептеулерді күтіп нәтижесін бір уақытта шығару нәтижелері

Мұндағы *GetValue* келесі есептеу есептелген алдыңғы нәтижені экранға шығармай блокқа қоятынын ескеріңіз.

Өздік жұмыс:

1. Күрделі итерациядан тұратын есептер таңдап, оған кодтың параллель орындалуы *Future* классының күту қызметі арқылы жүзеге асатын программа құрастырып нәтижесін салыстырыңыз.
2. Берілген мысалға есептеу уақытын көрсететін жолдар қосыңыз.

Бақылау сұрақтары:

1. Алдымен *Future* арқылы күту ұйымдастырылып, кейін кодтың

параллель орындалуын жіберуге болатындай RadStudio ортасында программа құруға бола ма?

2. Егер болатын болса көрсетіңіз?

Әдістемелік нұсқау

Теориялық материалдармен танысып, берілген жұмыстарды орындау.

Қолданылған әдебиеттер

Nick Hodges. Coding in RadStudio , Nepeta Enterprises 2014 - 2015

Тақырып 11. Сандық әдістерді параллель есептеу.

1. Сызықтық теңдеулер жүйесін параллель есептеу.
 2. Матрицаны-векторға жол бойынша бөліп көбейту
- Оқыту формасы: Компьютерлік практикум

Тапсырма 1:

n=50, 200, 400, 900 және worker-лардың әр түрлі санында тізбекті және параллельді матрица көбейтіндісі кодының орындалу уақытын анықтау (кестені толтыру).

N	Тізбектеп көбейту	Параллельді көбейту
50		
200		
400		
900		

Тапсырма 2:

Сықырлы шаршы түсініктемемін беру және түрлі n жағдайларына мысалдар келтіру.

Тапсырма 3:

Өз компьютерінің процессорының мінездемесін анықтап, тапсырмаларды орындау.

Өткізілу формасы: аудиториялық сабақ.

Ұсынылатын әдебиет: 1,2,6

Өткізілу формасы: компьютерлік практикум.

Әдістемелік нұсқаулар: практикалық сабақтың тақырыбына қарасты лекциялық материалды қарастырып, түсініктердің байланысын анықтау керек және пр ограммалау ортасында жүзеге асыру керек.

Тақырып 12. Сандық әдістерді параллель есептеу.

1. Анықталған интегралды параллель есептеу
 2. Параллель есептеу арқылы ауытқу шамаларын азайту
- Оқыту формасы: Компьютерлік практикум

Тапсырма 1:

Labindexпен numlabs функцияларының әртүрлі мәндерінде оң және сол тікбұрыштардың формулаларымен интегралды есептеңіз.

$$\int_{0.8}^{1.6} \frac{dx}{\sqrt{2x^2 + 1}};$$

Тапсырма 2: Сіздің кластеріңіздің сипатына орай трапеция формуласы арқылы келесі интегралды есептеңіз.

$$\int_{2,3}^{0,5} \frac{dx}{\sqrt{x^2 - 4}}$$

Өткізілу формасы: компьютерлік практикум.

Ұсынылатын әдебиет: 1,2,6

Әдістемелік нұсқаулар: практикалық сабақтың тақырыбына қарасты лекциялық материалды қарастырып, түсініктердің байланысын анықтау керек және программалау ортасында жүзеге асыру керек.

Тақырып 13. Сандық әдістерді параллель есептеу.

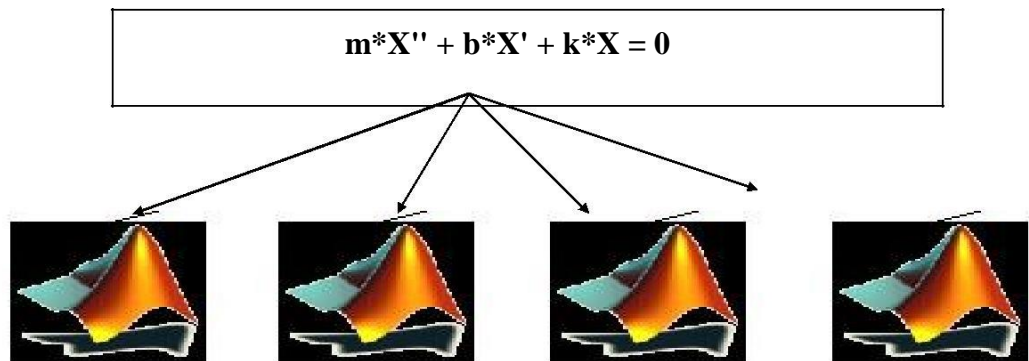
1. Дифференциальдық теңдеулерді параллель есептеу.

2. Параллель есептеу арқылы ауытқу шамаларын азайту

Оқыту формасы: Компьютерлік практикум

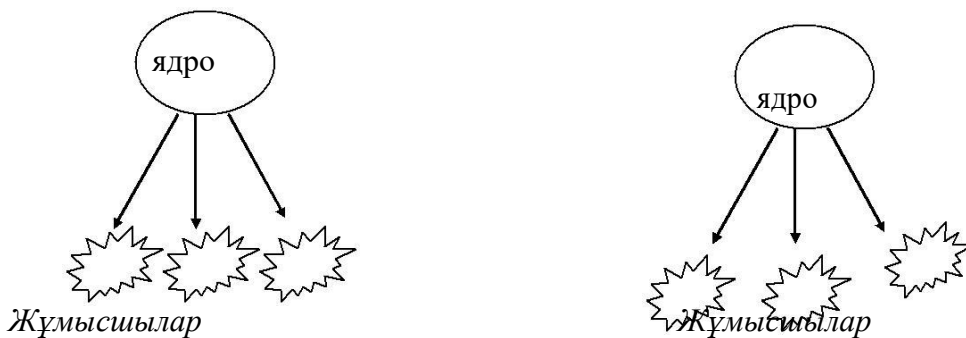
Тапсырма 1:

Берілген теңдеуді 4 worker негізінде орындау



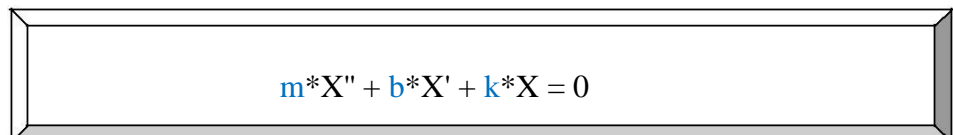
Тапсырма 2:

Берілген теңдеуді төменде көрсетілген ядро санына орай әртүрлі worker негізінде орындау



Тапсырма 3:

Берілген теңдеудегі мәндерді өзгертіп 4 worker негізінде орындау



Тапсырма 4:

Берілген мәліметтерді қолдана отырып, басқа қарапайым дифференциялық теңдеулерге бағдарлама жасау.

Өткізілу формасы: компьютерлік практикум.

Ұсынылатын әдебиет: 1,2,6

Әдістемелік нұсқаулар: практикалық сабақтың тақырыбына қарасты лекциялық материалды қарастырып, түсініктердің байланысын анықтау керек және программалау ортасында жүзеге асыру керек.

Тақырып 14. Оптимизация есептерін параллель есептеу.

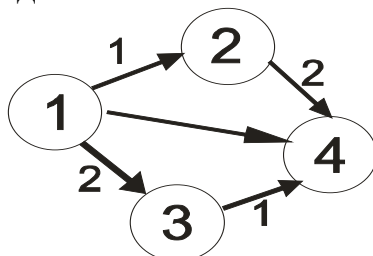
Мақсаты: Студенттерге ең кіші ара қашықтық туралы есепті параллель есептеп үйрету.

Кілттік сөздер: параллель есептеу, граф, ең кіші ара қашықтық

Оқытудың әдістемесі мен формасы: практикалық жұмыс

Бүтін санды программалау есебінің арасында көңіл бөлетін бір топ есептерді таңдап алуға болады. Бұл есептердің экстремумдық шешімін табу үшін берілген сандар жиынтығын алмастыру немесе комбинациялау арқылы түрлендіреміз. Осындай есептердің бірі ең кіші ара қашықтық туралы есебі. Осы мысалда ең кіші ара қашықтық есебін параллель есептеуге болатындығын көрсетеміз. Ол үшін алдымен ең кіші ара қашықтық туралы есебін түсіндіріп кетсек:

1. Граф дегеніміз төбелерді қосатын доғалар жиыны.
2. Бағытталған қырлары болатын графты тор деп атайды. Сонымен ең кіші ара қашықтық есебінде бір шығу нүктесі және бір жинау нүктесі болатын торды қарастырады.



Сурет 66—Граф

Егер осы тордың доғаларын ара қашықтық ретінде қарастырсақ, онда ең кіші ара қашықтық туралы есеп пайда болады.

Есептің шарты:

1. Тор түрінде беріледі.

2. $\left(i \begin{array}{c} C_{i,j} \\ j \end{array} \right)$ доғалар жиынтығымен беруге болады.

$$S : \{ 1 \stackrel{1}{=} 2, 1 \stackrel{2}{=} 3, 2 \stackrel{2}{=} 4, 3 \stackrel{1}{=} 4, 3 \stackrel{1}{=} 4, 1 \stackrel{4}{=} 4 \}$$

Ең кіші ара қашықтық есептің математикалық моделі:

$$F = \sum_{(i,j) \in S} \sum C_{i,j} \cdot X_j \rightarrow \min \quad (1)$$

$$X_{i,j} = 1, (i, j) \text{ тиімді матрица, } 0 - \text{ кері жағдай} \quad (2)$$

Сонымен (1) - (2) есебін шығару үшін келесі алгоритмді қолданамыз:

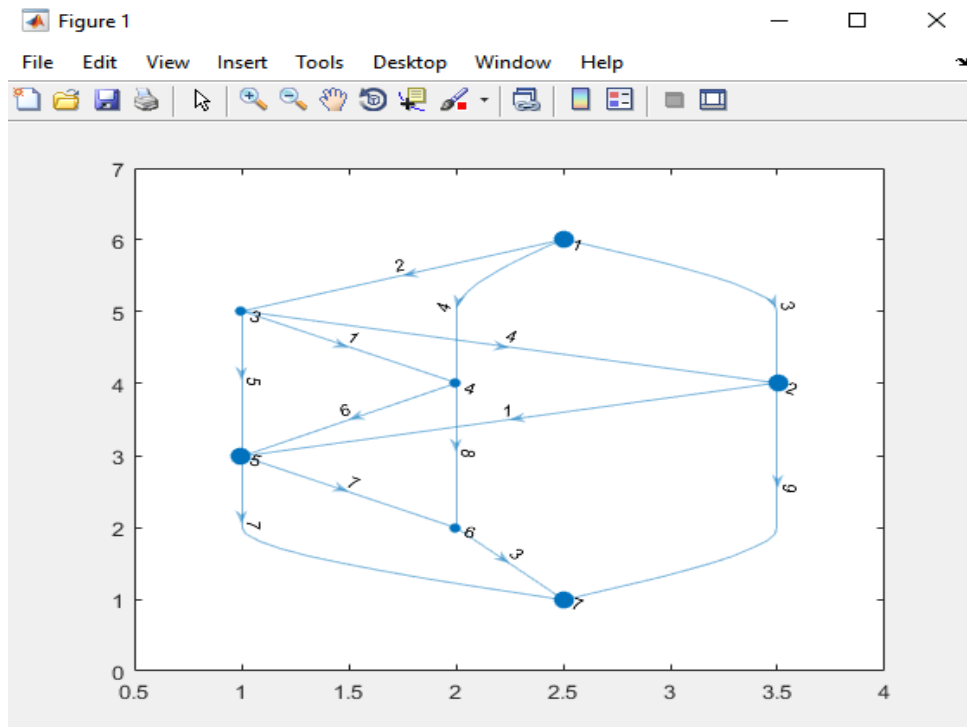
1. $y_1=0$ бір шығу нүктесі.
2. Кез-келген $y_1 = \min (y_i + C_{i,j})$ - ,барлық қалған $y_1 \quad j = \overline{2, n}$ - табамыз.
3. Табылған y_1 – мәндерін тордың сәйкес төбесіне қоямыз.
4. Егер $y_i - y_j = C_{i,j}$ болса, $\Rightarrow (i, j)$ доғаларды тиімді маршрутқа кіргіземіз.

Мысалы: 7 төбесі бар тор берілсін.

$(1-2), (2-7), (1-4), (3-4), (3-5), (4-5), (4-6), (5-7), (6-7), (1-3), (3-2), (2-5), (5-6)$

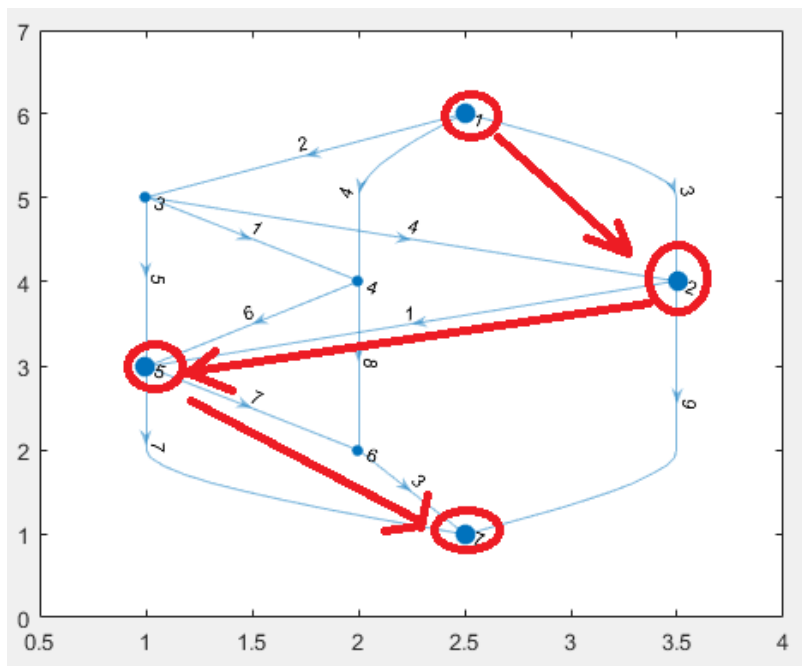
Бұл есепті параллель есептеп шығаруды көрсетіп тиімділігін анықтау үшін Matlab 2015 b жүйесінде алдымен тізбектей кейін параллель есептеу арқылы U шығу нүктесінен V нүктесіне дейінгі ең қысқа жолды табуға және нәтижесін графиктік түрғыда көруге болады. Есептеу Intel Core i5, 2.5GHz 4 ядролы процессорында тізбектей, кейін параллель есептелді.

Келесі суреттегі графика (67-сурет). Ең қысқа жолдары көрсетіледі.



Сурет 67- Ең қысқа ара қашықтық

Есептеу уақыттары әр түрлі, бірдей нәтижелер мен графиктертер алдық. Яғни, ең қысқа жол = 1 2 5 7 болса , жол ара қашықтығы = 11 болды.



Сурет 68- Ең қысқа ара қашықтық тиімді жолдары

Есептеу уақыттарын келесі кестеден көре аласыздар.

Кесте 7- Есептеу уақыттары

Тақырыбы	Есептеу уақыттары	Жылдамдату коэффициенті
Тізбектей есептеу	0,0724	
Параллель есептеу	0,0436	1,66

Осыдан ең кіші ара қашықтық есебін де параллель есептеуге болатындығын көре алдық. Түйіндері одан да көп есептер болса тиімділік одан да жақсы артатыны белгілі.

Тапсырма

Төмендегі интегралды өз варианттарыңыз бойынша таңдап алдымен local, кейін баптаған кластерлеріңізде есептеп нәтижесін салыстырыңыз.

- 1) $(1^2-2), (1^2-4), (1^4-3), (2^4-4), (2^1-5), (4^1-3), (4^6-5), (3^3-5).$
- 2) $(5^{10}-2), (5^{15}-4), (5^{15}-6), (6^{10}-4), (6^{20}-3), (4^{10}-1), (3^{20}-4), (3^{15}-1), (2^{10}-4), (2^{30}-1).$
- 3) $(1^2-2), (1^7-4), (1^4-5), (1^1-3), (2^5-4), (3^2-5), (3^2-6), (5^2-4), (6^1-5), (6^3-4).$
- 4) $(2^2-1), (2^3-5), (2^3-6), (3^3-4), (3^4-6), (1^1-6), (1^6-4), (5^4-3), (5^2-6), (6^2-4).$

5) $(3-1)^2, (3-2)^3, (3-4)^4, (3-6)^6, (1-6)^5, (2-4)^2, (2-5)^2, (4-6)^2, (5-4)^1, (5-6)^1$.

Бақылау сұрақтары

1. Ең қысқа ара қашықтық есебі параллель есептеу әр кезде тиімді бола ма?
2. Берілген мысалда есептеудің жедел орындалуы қандай шамамен есептеледі?

Тақырып 15. GPU (Graphics Processing Unit) қосымша есептеу қуаты ретінде пайдалану.

3. GPU да есептеу
 4. GPU да параллель есептеу және салыстыру
- Оқыту формасы: Компьютерлік практикум

Тапсырма 1: Мандельброт жиынтығының 4 түрлі жүзеге асырылуын қарастыру

Тапсырма 2: Жюлио жиынтығын ядролардың әртүрлі санында есептеу және функцияларды қолдану жағдайына қарай нәтижелерін салыстыру

Өткізілу формасы: компьютерлік практикум.

Ұсынылатын әдебиет: 1,2,6,11

Әдістемелік нұсқаулар: практикалық сабақтың тақырыбына қарасты лекциялық материалды қарастырып, түсініктердің байланысын анықтау керек және программалау ортасында жүзеге асыру керек.