



عملي – البرمجة التفرعية

الطلاب :

رانيا الغزالي – سارة شريفة

محمد أبي دبا – مرح عثمان

الإشراف:

م.روان قرعوني

لدينا مركز توزيع أدوية لكل صيدليات وعيادات الأطباء في محافظة دمشق  
ونريد كتابة برنامج تفرعي لإدارة عمل موزعي المركز على التوازي  
بحيث يتم تخديم أكثر صيدلية وعيادة طبيب في نفس الوقت من قبل  
الموزعين

سنقوم باقتراح برنامج تفرعي لحل مسألة توزيع حيث يمثل البرنامج التالي  
مركز توزيع الأدوية

### السيناريو التفرعي:

قمنا باستعمال بيئة MPI لحل المسألة المطروحة

حيث قمنا بتقسيم البرنامج التفرعي الى مستويين

المستوى الاول يعبر عن موزعي الأدوية

المستوى الثاني يعبر عن عن اهداف التوزيع (الصيدليات والعيادات)

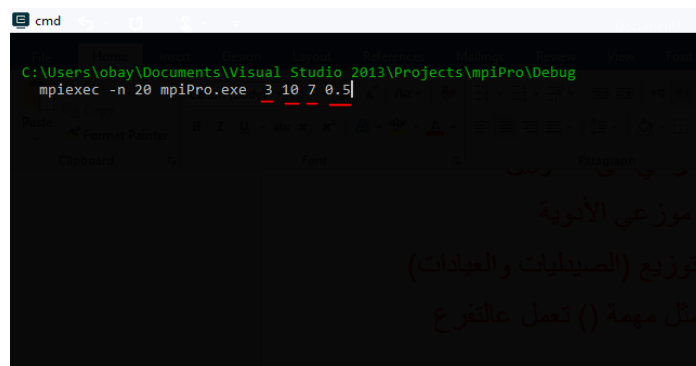
كل مستوى من المستويات السابقة يمثل مهمة ( ) تعمل عالتفرع

## وسطاء البرنامج (الدخل):

- عدد الموزعين
- عدد الصيدليات
- عدد عيادات الأطباء
- الزمن الوسطي لتنفيذ عملية توزيع

```
int p1, //number of distrubitors  
    p2, //number of pharmacies  
    p3, //number of clinics  
    p4; // mean time
```

يتم تمرير الوسطاء عبر الـ (command line) بشكل عام لكل نسخ البرنامج:



```
cmd  
C:\Users\obay\Documents\Visual Studio 2013\Projects\mpiPro\Debug  
mpirexec -n 20 mpiPro.exe 3 10 7 0.5
```

موزعي الأدوية  
توزيع (الصيدليات والعيادات)  
مثل مهمة ( ) تعمل على توزيع

## المتحولات المستخدمة:

```
int rank, // current task rank in the MPI_COMM_WORLD
    tasks, // number of tasks in MPI_COMM_WORLD
    rc;    // variable used for confirming send & receive operation
int mod, source;
int tag = 9;
char in, done = 't';
MPI_Status stat;
```

- Rank: يمثل رقم المهمة الحالية
- Tasks يمثل عدد نسخ البرنامج
- Rc للتحقق من نجاح عمليات الارسال والاستقبال
- Mod & source يستعملان في تقسيم المواقع على الموزعين
- In & done يستخدمان للارسال والاستقبال
- Stat يستخدم في عملية الارسال
- Tag للارسال و الاستقبال

```
if (argc == 5){
    p1 = atoi(argv[1]);
    p2 = atoi(argv[2]);
    p3 = atoi(argv[3]);

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &tasks);
```

قراءة المتحولات من ال (command line) وتحويلها الى الانماط

المناسبة

القيام بتهيئة بروتوكول ال mpi و الحصول على معلومات فضاء

الاتصال

والحصول على رقم المهمة في فضاء الاتصال

## مستويات التفرع:

### • مستوى الموزعين:

نقوم بتقسيم النقاط المراد الارسال اليها على الموزعين باخذ باقي  
قسمة ال() على عدد الموزعين حيث يكون الباقي هو رقم الموزع  
المسؤول ومن ثم نقوم بعملية الارسال مع طباعة اشعار للعملية  
(مستقبل ,مرسل)

```
if (rank < p1){  
    for (int i = p1; i < tasks; i++){  
        mod = i%p1;  
        if (rank == mod){  
            rc = MPI_Send(&done, 1, MPI_CHAR, i, tag, MPI_COMM_WORLD);  
  
            if (rc == MPI_SUCCESS)  
                cout << "Provider id : " << rank << " Sent a Delivery to location : " << i-p1 << endl;  
        }  
    }  
}
```

### • مستوى المستقبلين (عيادات - صيدليات):

بنفس الاسلوب نحدد (rank) المرسل ونقوم بعملية الاستقبال  
مع طباعة اشعار بتحقق عملية الاستلام في حال كان  
(صيدلية أو عيادة)

```
else{//delivery  
    source = rank %p1;  
    rc = MPI_Recv(&in, 1, MPI_CHAR, source, tag, MPI_COMM_WORLD, &stat);  
    if (rc == MPI_SUCCESS){  
        if(rank < p1+p2)  
            cout << "pharmacy : " << rank - p1 << " Recived a delivery from provider : " << source << endl;  
        else{  
            cout << "clinic : " << rank - p1 - p2 << " Recived a delivery from provider : " << source << endl;  
        }  
    }  
}
```

وبعد القيام بتنفيذ البرنامج عالمستويين السابقين نقوم بانتهاء عمل  
برتوكول MPI  
وطباعة الزمن الذي استغرقه تنفيذ البرنامج

```
MPI_Finalize();  
  
cout << "time of execution is : " << (double)(clock() - tStart) / CLOCKS_PER_SEC;
```

في حال عدم ادخال الوسطاء بشكل صحيح او ناقص  
يقوم البرنامج بطباعة اشعار بالخطأ

```
else{  
    cout << "Errorr! bad input ";  
    MPI_Abort(MPI_COMM_WORLD,rc);  
}
```

## امثلة تنفيذ

[illegible]

```
C:\Users\obay\Documents\Visual Studio 2013\Projects\mpiPro\Debug
mpirun -n 10 mpiPro.exe 2 4 4 0.35
Provider id :1 Sent a Delivery to location :1
Provider id :0 Sent a Delivery to location :0
pharmacy :0 Recived a delivery from provider :0
Provider id :1 Sent a Delivery to location :3
pharmacy :1 Recived a delivery from provider :1
Provider id :0 Sent a Delivery to location :2
Provider id :1 Sent a Delivery to location :5
Provider id :0 Sent a Delivery to location :4
pharmacy :3 Recived a delivery from provider :1
Provider id :1 Sent a Delivery to location :7
pharmacy :2 Recived a delivery from provider :0
clinic :1 Recived a delivery from provider :1
Provider id :0 Sent a Delivery to location :6
clinic :0 Recived a delivery from provider :0
clinic :3 Recived a delivery from provider :1
clinic :2 Recived a delivery from provider :0
```

```
task # 7 time of execution is : 0.237
task # 9 time of execution is : 0.44
task # 5 time of execution is : 0.284
task # 2 time of execution is : 0.161
task # 4 time of execution is : 0.286
task # 6 time of execution is : 0.239
task # 8 time of execution is : 0.286
task # 10 time of execution is : 0.412
task # 3 time of execution is : 0.252
task # 11 time of execution is : 0.413
task # 13 time of execution is : 0.475
task # 1 time of execution is : 0.195
task # 0 time of execution is : 0.132
task # 12 time of execution is : 0.46
task # 14 time of execution is : 0.414
```

**نلاحظ** ان زمن التنفيذ التفرعي (زمن تنفيذ اطول عملية)  
في هذه الحالة يساوي (0.475) وهو زمن صغير جدا مقارنة بتنفيذ  
البرنامج تسلسليا حيث يمكن ان يصل زمن تنفيذ برنامج من الشكل  
السابق الى 4 ثواني  
ويزداد الوقت بنقصان عدد الموزعين

```
task # 1 time of execution is : 0.218
task # 2 time of execution is : 0.218
task # 3 time of execution is : 0.257
task # 4 time of execution is : 0.249
task # 5 time of execution is : 0.14
task # 7 time of execution is : 0.518
task # 6 time of execution is : 0.156
task # 8 time of execution is : 0.14
task # 9 time of execution is : 0.408
task # 10 time of execution is : 0.156
task # 11 time of execution is : 0.361
task # 13 time of execution is : 0.501
task # 14 time of execution is : 0.423
task # 12 time of execution is : 0.533
task # 15 time of execution is : 0.408
task # 16 time of execution is : 0.423
task # 17 time of execution is : 0.533
task # 18 time of execution is : 0.329
task # 19 time of execution is : 0.486
task # 0 time of execution is : 0.273
```