

بنام خدا

## تمرین MPI سیستم‌های توزیعی

میلاد تیموری ۹۵۷۲۵۱۲۷

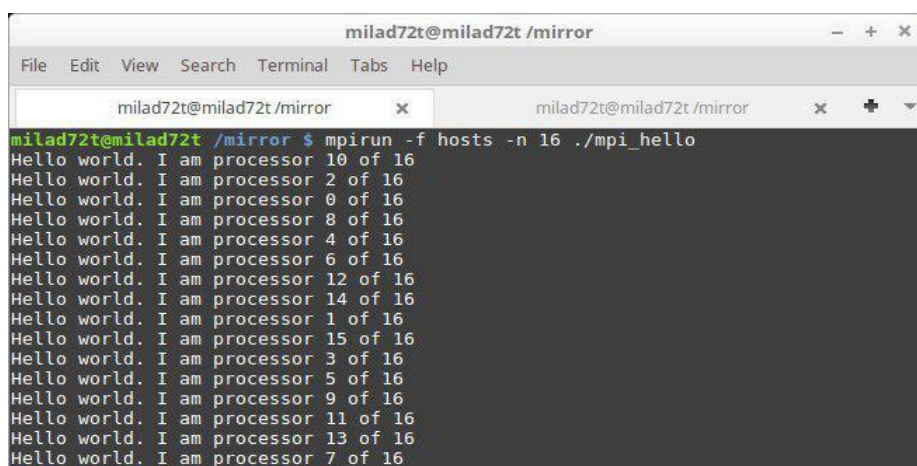
دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران

پست الکترونیکی: [milad72t@gmail.com](mailto:milad72t@gmail.com)

آبان ۹۶

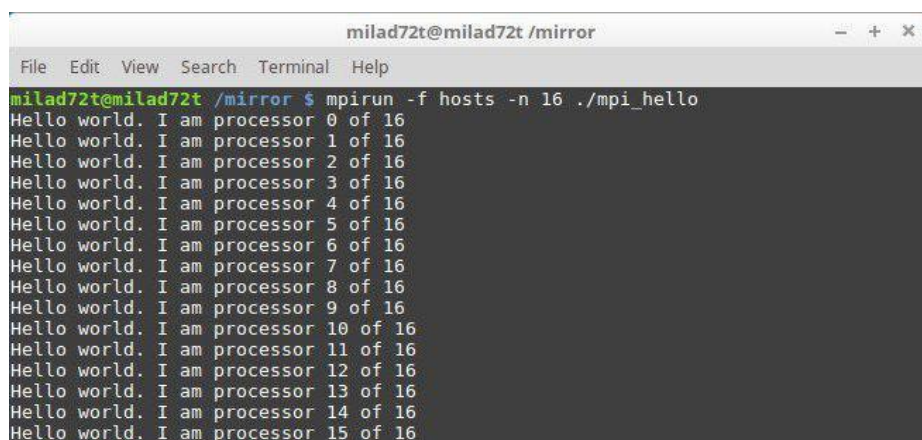
Hello World موازی

(a) ترتیب اجرای فرآیندها و به دنبال آن اجرای خط چاپ Hello Word بر روی خروجی، هیچ ترتیبی ندارد و به هر ترتیبی ممکن است عبارت بر روی صفحه خروجی ظاهر شود



```
milad72t@milad72t /mirror
File Edit View Search Terminal Tabs Help
milad72t@milad72t /mirror
milad72t@milad72t /mirror $ mpirun -f hosts -n 16 ./mpi_hello
Hello world. I am processor 10 of 16
Hello world. I am processor 2 of 16
Hello world. I am processor 0 of 16
Hello world. I am processor 8 of 16
Hello world. I am processor 4 of 16
Hello world. I am processor 6 of 16
Hello world. I am processor 12 of 16
Hello world. I am processor 14 of 16
Hello world. I am processor 1 of 16
Hello world. I am processor 15 of 16
Hello world. I am processor 3 of 16
Hello world. I am processor 5 of 16
Hello world. I am processor 9 of 16
Hello world. I am processor 11 of 16
Hello world. I am processor 13 of 16
Hello world. I am processor 7 of 16
```

(b) برای اینکار نیاز به یک هماهنگ کننده است، هر برنامه rank خود را به root میفرستد و فرآینده root به ترتیب این مقادیر را از فرآیندهای دیگر دریافت میکند و بر روی صفحه خروجی نمایش میدهد.



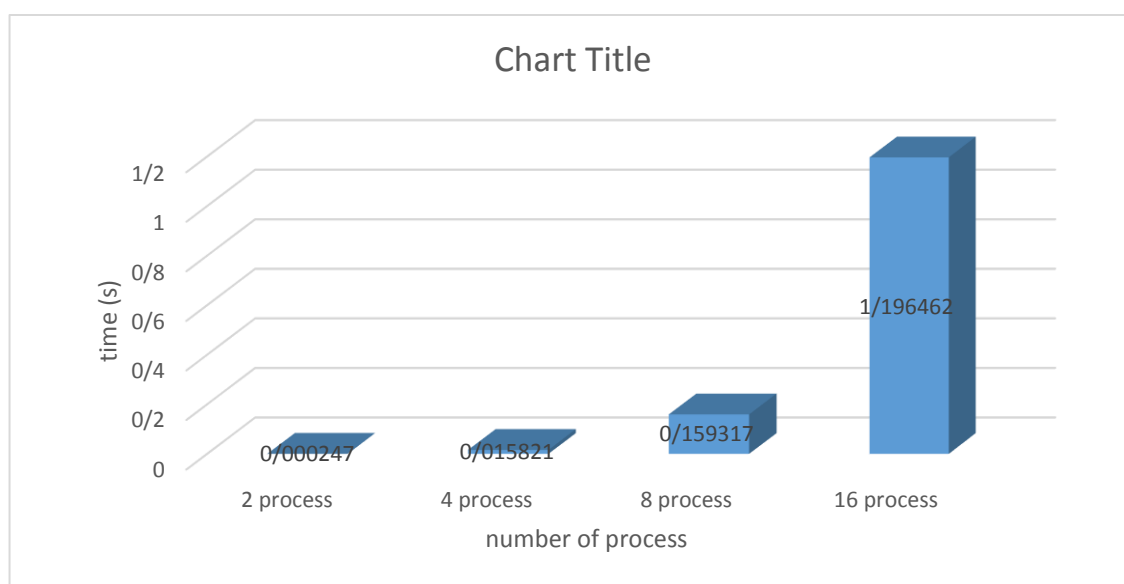
```
milad72t@milad72t /mirror
File Edit View Search Terminal Help
milad72t@milad72t /mirror $ mpirun -f hosts -n 16 ./mpi_hello
Hello world. I am processor 0 of 16
Hello world. I am processor 1 of 16
Hello world. I am processor 2 of 16
Hello world. I am processor 3 of 16
Hello world. I am processor 4 of 16
Hello world. I am processor 5 of 16
Hello world. I am processor 6 of 16
Hello world. I am processor 7 of 16
Hello world. I am processor 8 of 16
Hello world. I am processor 9 of 16
Hello world. I am processor 10 of 16
Hello world. I am processor 11 of 16
Hello world. I am processor 12 of 16
Hello world. I am processor 13 of 16
Hello world. I am processor 14 of 16
Hello world. I am processor 15 of 16
```

```

milad72t@milad72t /mirror
File Edit View Search Terminal Tabs Help
milad72t@milad72t /mirror x milad72t@milad72t /mirror x +
milad72t@milad72t /mirror $ mpirun -f hosts -n 16 ./mpi_calcPi 12000000
in Quarter count: 9423977 from : 12000000 count By process 4
in Quarter count: 9424088 from : 12000000 count By process 6
in Quarter count: 9424976 from : 12000000 count By process 10
in Quarter count: 9424780 from : 12000000 count By process 14
in Quarter count: 9424226 from : 12000000 count By process 2
in Quarter count: 9424465 from : 12000000 count By process 8
in Quarter count: 9425739 from : 12000000 count By process 12
in Quarter count: 9424176 from : 12000000 count By process 13
in Quarter count: 9424282 from : 12000000 count By process 15
in Quarter count: 9423932 from : 12000000 count By process 1
in Quarter count: 9426508 from : 12000000 count By process 5
in Quarter count: 9423183 from : 12000000 count By process 9
in Quarter count: 9422709 from : 12000000 count By process 11
in Quarter count: 9423902 from : 12000000 count By process 7
in Quarter count: 9426185 from : 12000000 count By process 3
pi : 3.1414917333

```

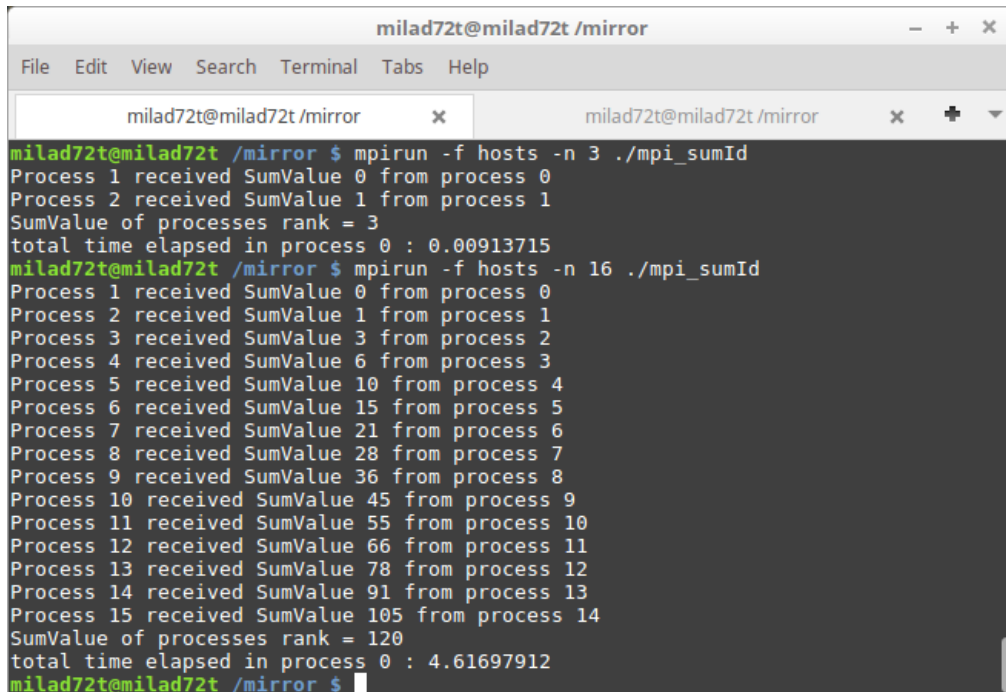
در برنامه نوشته شده به هر فرآیند تعداد دفعاتی که برای محاسبه عدد پی باید انجام شود، داده میشود. به عنوان مثال در برنامه فوق ۱۶ فرآیند هرکدام ۱۲ میلیون بار اقدام به تخمین عدد پی کرده اند.



تعداد فرآیند	۲ فرآیند	۴ فرآیند	۸ فرآیند	۱۶ فرآیند
عدد پی	۳,۱۴۲۲۱۰	۳,۱۴۱۹۲۰	۳,۱۴۱۶۱۶	۳,۱۴۱۴۹۱

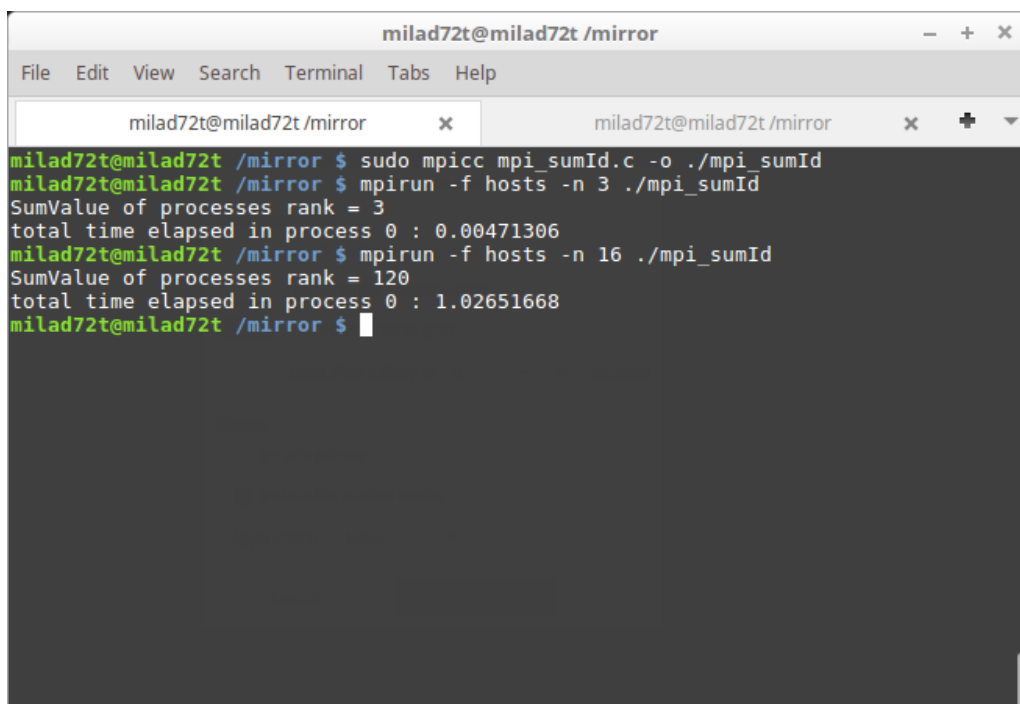
جمع rank فرآیندها

با استفاده از توابع send و receive



```
milad72t@milad72t /mirror
File Edit View Search Terminal Tabs Help
milad72t@milad72t /mirror x milad72t@milad72t /mirror x + v
milad72t@milad72t /mirror $ mpirun -f hosts -n 3 ./mpi_sumId
Process 1 received SumValue 0 from process 0
Process 2 received SumValue 1 from process 1
SumValue of processes rank = 3
total time elapsed in process 0 : 0.00913715
milad72t@milad72t /mirror $ mpirun -f hosts -n 16 ./mpi_sumId
Process 1 received SumValue 0 from process 0
Process 2 received SumValue 1 from process 1
Process 3 received SumValue 3 from process 2
Process 4 received SumValue 6 from process 3
Process 5 received SumValue 10 from process 4
Process 6 received SumValue 15 from process 5
Process 7 received SumValue 21 from process 6
Process 8 received SumValue 28 from process 7
Process 9 received SumValue 36 from process 8
Process 10 received SumValue 45 from process 9
Process 11 received SumValue 55 from process 10
Process 12 received SumValue 66 from process 11
Process 13 received SumValue 78 from process 12
Process 14 received SumValue 91 from process 13
Process 15 received SumValue 105 from process 14
SumValue of processes rank = 120
total time elapsed in process 0 : 4.61697912
milad72t@milad72t /mirror $
```

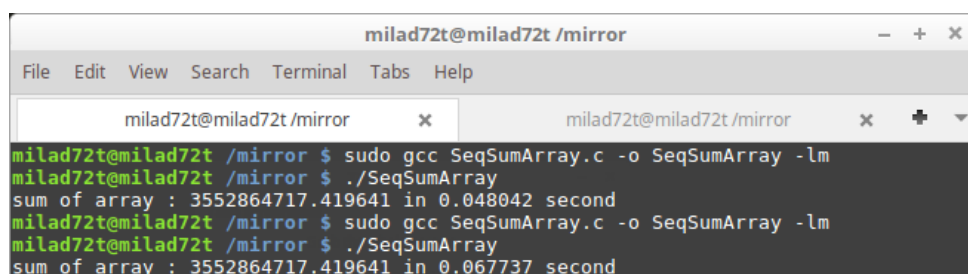
با استفاده از تابع AllReduce



```
milad72t@milad72t /mirror
File Edit View Search Terminal Tabs Help
milad72t@milad72t /mirror x milad72t@milad72t /mirror x + v
milad72t@milad72t /mirror $ sudo mpicc mpi_sumId.c -o ./mpi_sumId
milad72t@milad72t /mirror $ mpirun -f hosts -n 3 ./mpi_sumId
SumValue of processes rank = 3
total time elapsed in process 0 : 0.00471306
milad72t@milad72t /mirror $ mpirun -f hosts -n 16 ./mpi_sumId
SumValue of processes rank = 120
total time elapsed in process 0 : 1.02651668
milad72t@milad72t /mirror $
```

## پردازش آرایه

در کد سریال، زمانی که حلقه اصلی بر روی سطر باشد زمان کمتری صرف پردازش آرایه میشود، زیرا آرایه در حافظه به صورت ترتیبی ذخیره میشوند و زمانی که حلقه اصلی بر روی سطر آرایه زده شود، داده های آرایه به صورت ترتیبی فراخوانی میشوند و در این حالت کش میتواند عملکرد بهتری داشته باشد.



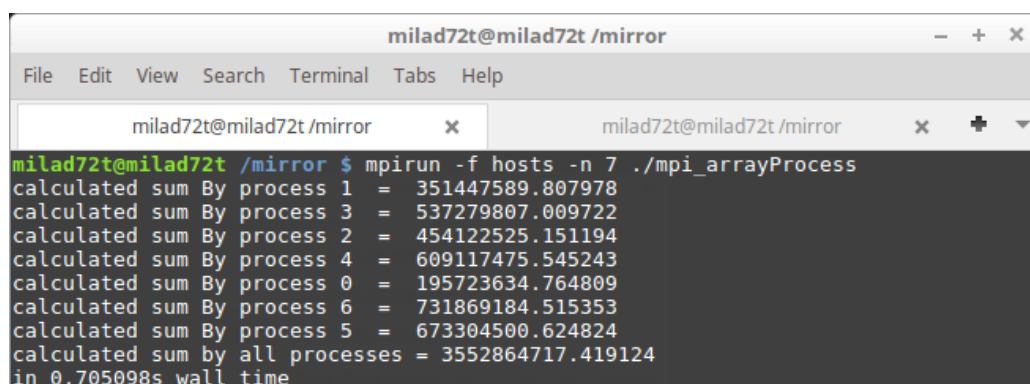
```
milad72t@milad72t /mirror
File Edit View Search Terminal Tabs Help
milad72t@milad72t /mirror x milad72t@milad72t /mirror x + v
milad72t@milad72t /mirror $ sudo gcc SeqSumArray.c -o SeqSumArray -lm
milad72t@milad72t /mirror $ ./SeqSumArray
sum of array : 3552864717.419641 in 0.048042 second
milad72t@milad72t /mirror $ sudo gcc SeqSumArray.c -o SeqSumArray -lm
milad72t@milad72t /mirror $ ./SeqSumArray
sum of array : 3552864717.419641 in 0.067737 second
```

در حالت موازی master با دانستن بعد آرایه و تعداد فرآیندها اقدام به محاسبه تعداد چانکی که به هر آرایه باید فرستاده شود میکند:

$$\text{chunk} = N / \text{NumProcess}$$

از آنجایی که پاسخ این تقسیم همواره عدد صحیح نیست ممکن است چند سطر از آرایه به هیچ فرآیندی تخصیص نیابد بنابراین master با محاسبه این مقدار از سطرها، خود علاوه بر مقدار چانک محاسبات این سطرهای باقیمانده را نیز انجام میدهد:

$$\text{remaining} = N \% \text{NumProcess};$$



```
milad72t@milad72t /mirror
File Edit View Search Terminal Tabs Help
milad72t@milad72t /mirror x milad72t@milad72t /mirror x + v
milad72t@milad72t /mirror $ mpirun -f hosts -n 7 ./mpi_arrayProcess
calculated sum By process 1 = 351447589.807978
calculated sum By process 3 = 537279807.009722
calculated sum By process 2 = 454122525.151194
calculated sum By process 4 = 609117475.545243
calculated sum By process 0 = 195723634.764809
calculated sum By process 6 = 731869184.515353
calculated sum By process 5 = 673304500.624824
calculated sum by all processes = 3552864717.419124
in 0.705098s wall time
```