

## Keskkondade paigaldamine

Järgnevas õpetuses näidatakse kuidas paigaldada MPI ja Charm++ keskkonda linux masinale. Õpetus on testitud Ubuntu operatsioonisüsteemi peal (täpsemalt „ubuntu-10.04.2-desktop-i386“) ja täpselt selles järjekorras käivitati koodi. Kui soovitakse paigaldada ennem MPI standard, siis uuendus ja ssh tuleb eelevalt paigaldada. Järgnevas dokumendis eeldatakse, et masin on internetiga ühenduses.

## Charm++ raamistiku paigaldamine

1. Kõige alguses tuleks veenduda kas linux on uuendatud:

```
sudo apt-get update
```

2. Esmalt tuleb tõmmata endale charm++ „binaries“ fail:

```
charm-6.2.1_multicore-linux64_production.tar.gz
```

Antud töös kasutatakse Charm++ 6.2.1 stabiilset versiooni leheküljelt:

```
http://charm.cs.uiuc.edu/download/
```

3. Saadud fail tuleks lahti pakkida ja järgnevals installida vajalik tööriist, et selle sisu installida:

```
sudo apt-get install build-essential
```

4. Edasi minna lahtipakitud kausta sisse:

```
.../charm-6.2-multicore-linux64
```

5. Ja ehitada endale keskkond:

```
./build
```

6. Küsitakse:

- Do you have a special network interconnect?  
✓ N
- How do you want to handle SMP/Multicore: [1-4]
  - 1) Single-threaded [default]
  - 2) Multicore(single node only)
  - 3) SMP
  - 4) POSIX Shared Memory✓ 1
- Do you want to specify a compiler?  
✓ N

- Do you want to specify any Charm++ build options, such as fortran compilers?  
✓ N
  - Choose a set of compiler flags [1-5]
    - 1) None
    - 2) Debug
    - 3) Optimized [default]
    - 4) Optimized no projections
    - 5) Custom
 ✓ 1
  - What do you want to build?
    - 1) Charm++ [default] (choose this if you are building NAMD)
    - 2) Charm++, AMPI, ParFUM, FEM and other libraries
 ✓ 1
  - Do you want to compile in parallel?
    - 1) No
    - 2) Build with -j2
    - 3) Build with -j4
    - 4) Build with -j8
    - 5) Build with -j16 [default]
    - 6) Build with -j32
    - 7) Build with -j
 ✓ 1
- We have determined a suitable build line is:
- ```
./build charm++ net-linux
```
- Do you want to start the build now?  
✓ Y

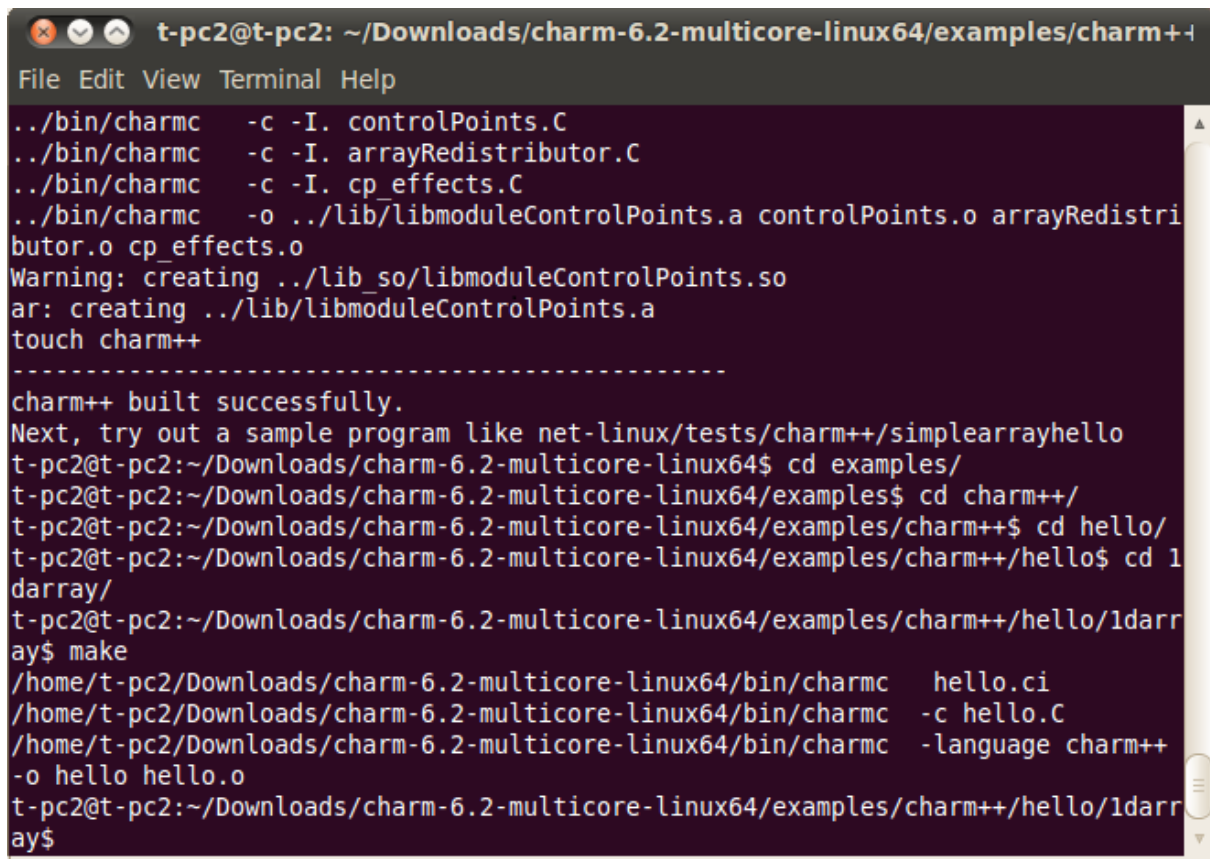
Olenevalt masina kiirusele võtab ehitamine keskmiselt 2 - 7 minutit aega, kui kõik vajalikud failid luuakse ja raamistik paigaldatakse

7. Et testida kas senini on kõik korralikult läinud, tuleks minna kausta:

```
./charm-6.2-multicore-linux64/examples/charm++/hello/1darray
```

Ja käivitada käsk:

```
make
```

A terminal window titled 't-pc2@t-pc2: ~/Downloads/charm-6.2-multicore-linux64/examples/charm++'. The window shows the execution of 'charm++' to build a sample program. It starts by compiling three C files (controlPoints.C, arrayRedistributor.C, cp\_effects.C) into object files, then linking them into a static library 'libmoduleControlPoints.a' and a shared library 'libmoduleControlPoints.so'. A warning is shown about creating the shared library. The process then builds 'charm++' successfully. The user navigates through the directory structure: 'examples/' to 'charm++/' to 'hello/' to 'ldarray/'. Finally, the user runs 'make', which compiles 'hello.C' into 'hello.ci', then 'hello.ci' into 'hello.o' using 'charm++' as the compiler, and finally links 'hello.o' into the executable 'hello'.

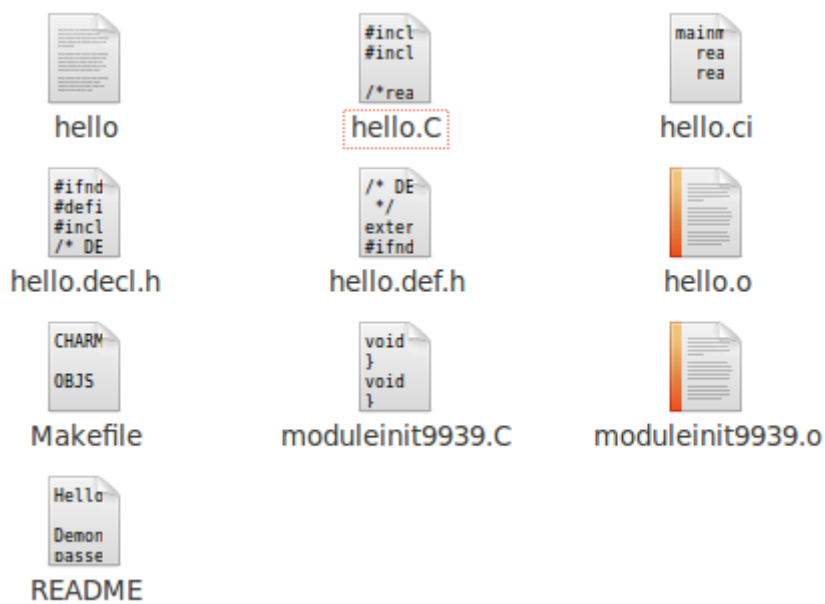
Joonis 1. Väljalõige edukast Charm++ raamistiku paigaldamisest.

8. Kui „make“ õnnestus ja kuvatakse umbes sarnane väljund nagu eelneval joonisel, siis võib järgmisena kompileeritud faili käivitada.

Kui tähele panna siis alguses oli seal kaustas vaid neli faili:

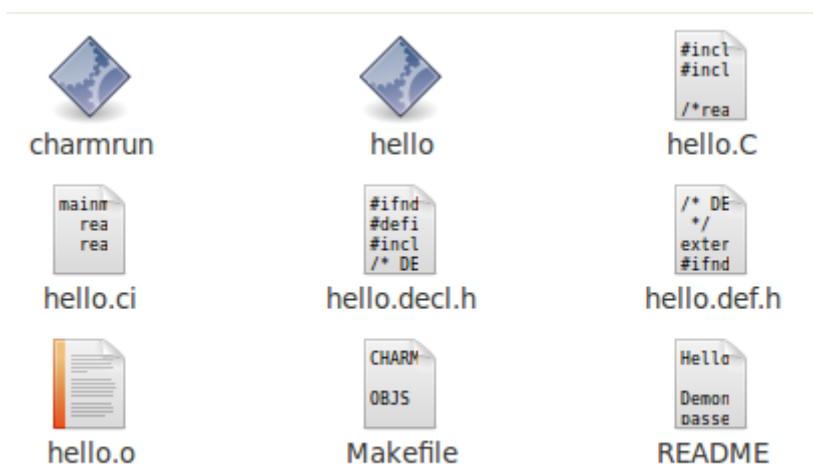
- Hello.C
- Hello.ci
- Makefile
- README

Kuid pärast kompileerimist on ajutiselt rohkem faile:



Joonis 2. Ajutiselt tekkinud failid.

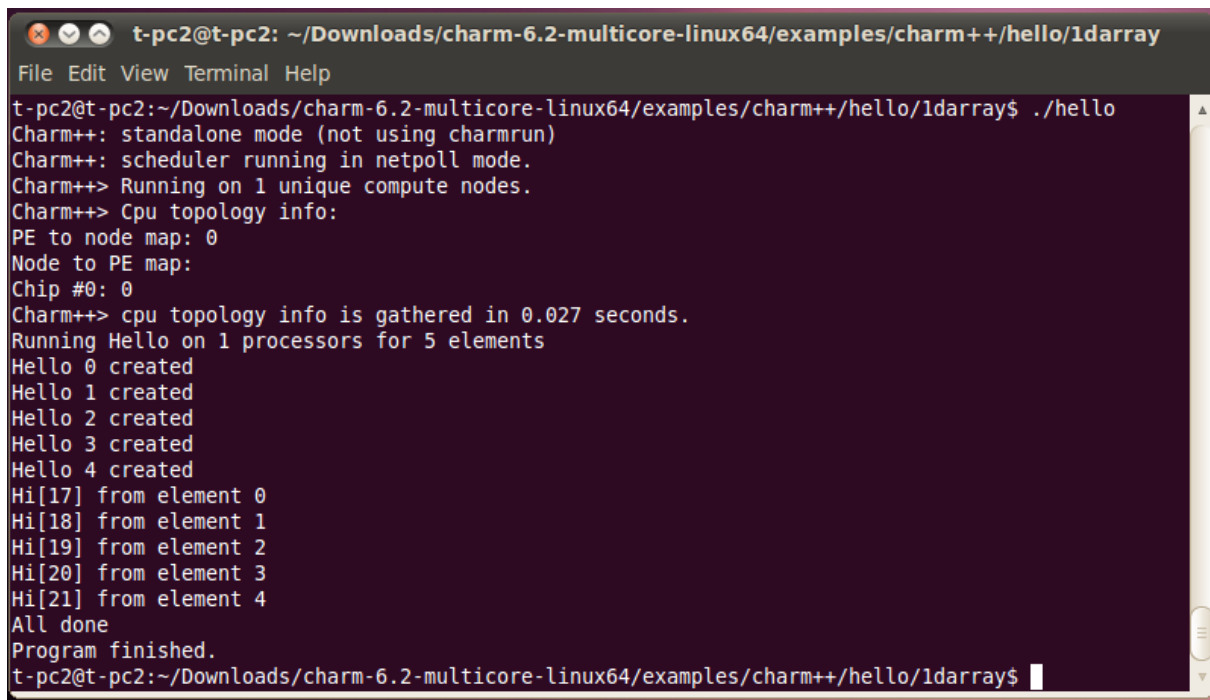
Ja pärast mõnda hetke on juba käivitamiseks vajalikud failid olemas ja mooduli failid kustutatud:



Joonis 3. Kompileeritud faile sisaldava kausta väljalõige.

9. Järgmiseks käivitada programm ühe protsessori peal:

./hello



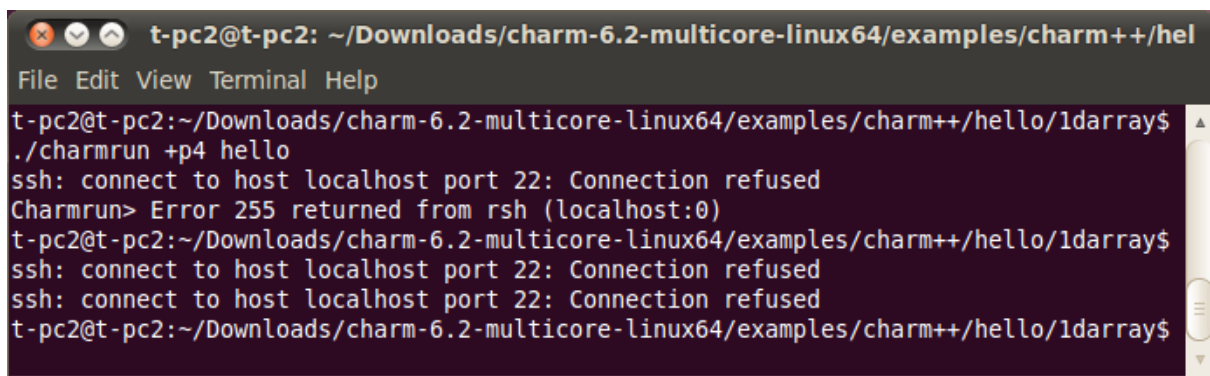
```
t-pc2@t-pc2: ~/Downloads/charm-6.2-multicore-linux64/examples/charm++/hello/1darray
File Edit View Terminal Help
t-pc2@t-pc2:~/Downloads/charm-6.2-multicore-linux64/examples/charm++/hello/1darray$ ./hello
Charm++: standalone mode (not using charmrun)
Charm++: scheduler running in netpoll mode.
Charm++> Running on 1 unique compute nodes.
Charm++> Cpu topology info:
PE to node map: 0
Node to PE map:
Chip #0: 0
Charm++> cpu topology info is gathered in 0.027 seconds.
Running Hello on 1 processors for 5 elements
Hello 0 created
Hello 1 created
Hello 2 created
Hello 3 created
Hello 4 created
Hi[17] from element 0
Hi[18] from element 1
Hi[19] from element 2
Hi[20] from element 3
Hi[21] from element 4
All done
Program finished.
t-pc2@t-pc2:~/Downloads/charm-6.2-multicore-linux64/examples/charm++/hello/1darray$
```

Joonis 4. Charm++ kood töötab ühe protsessori peal.

Kui nüüd kohe proovida paralleelselt jooksutada käsuga:

```
./charmrun +p4 hello
```

siis tekib umbes selline veateade:



```
t-pc2@t-pc2: ~/Downloads/charm-6.2-multicore-linux64/examples/charm++/hel
File Edit View Terminal Help
t-pc2@t-pc2:~/Downloads/charm-6.2-multicore-linux64/examples/charm++/hello/1darray$ ./charmrun +p4 hello
ssh: connect to host localhost port 22: Connection refused
Charmrun> Error 255 returned from rsh (localhost:0)
t-pc2@t-pc2:~/Downloads/charm-6.2-multicore-linux64/examples/charm++/hello/1darray$
ssh: connect to host localhost port 22: Connection refused
ssh: connect to host localhost port 22: Connection refused
t-pc2@t-pc2:~/Downloads/charm-6.2-multicore-linux64/examples/charm++/hello/1darray$
```

Joonis 5. Mitme protsessori peal jooksutamise veateade.

10. Esmalt tuleb kindlaks teha kas ssh klient ja server on paigaldatud, kui pole siis:

```
sudo apt-get install openssh-server openssh-client
```

11. Et jooksutada charm programme jagatud ressurssidel, tuleb esmalt genereerida võti:

```
ssh-keygen
```

Generating public/private rsa key pair.

Enter file in which to save the key (/home/t-pc/.ssh/id\_rsa):

Jätta tühjaks ja vajutada enter.

Kui võti on juba loodud:

/home/t-pc/.ssh/id\_rsa already exists.

Overwrite (y/n)?

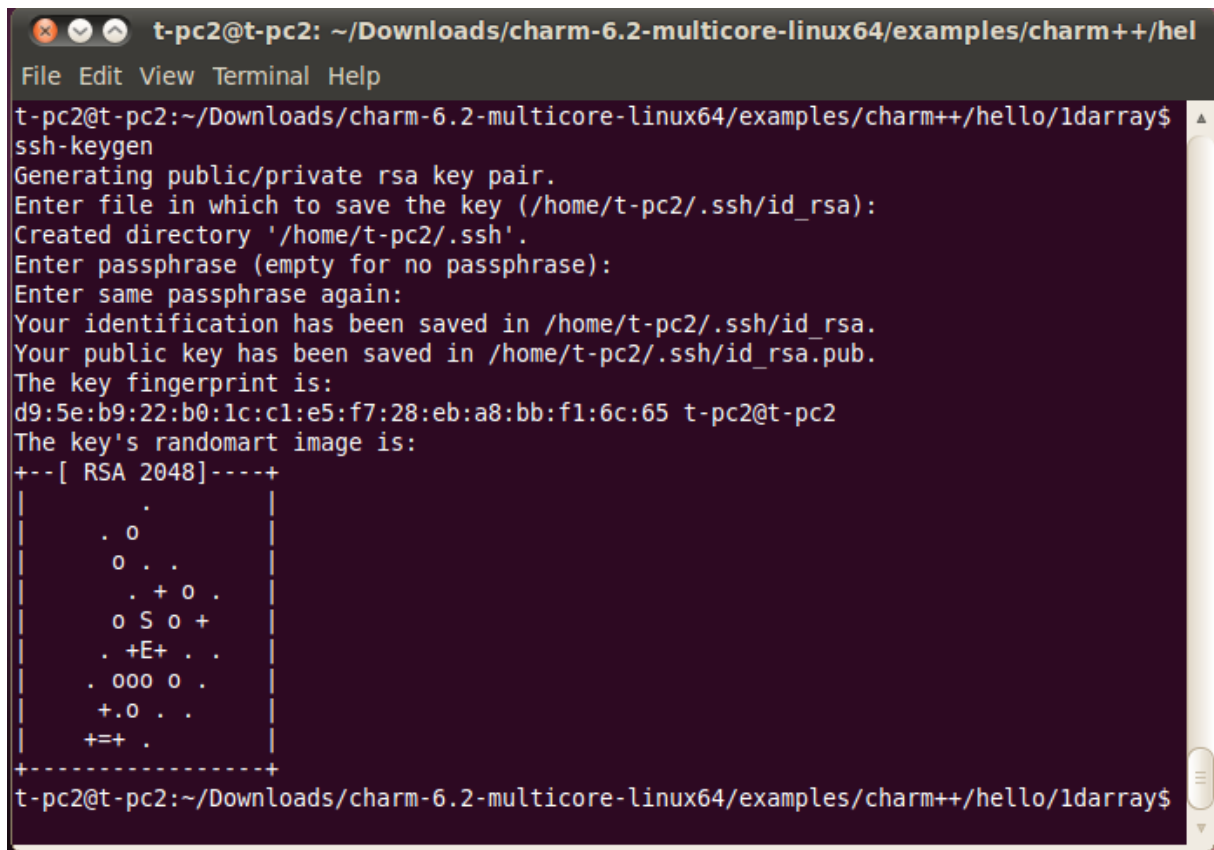
Vajutada y ja enter.

Enter passphrase (empty for no passphrase):

Jätta tühjaks ja vajutada enter.

Enter same passphrase again:

Jätta tühjaks ja vajutada enter.



```
t-pc2@t-pc2: ~/Downloads/charm-6.2-multicore-linux64/examples/charm++/hel
File Edit View Terminal Help
t-pc2@t-pc2:~/Downloads/charm-6.2-multicore-linux64/examples/charm++/hello/ldarray$
ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/t-pc2/.ssh/id_rsa):
Created directory '/home/t-pc2/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/t-pc2/.ssh/id_rsa.
Your public key has been saved in /home/t-pc2/.ssh/id_rsa.pub.
The key fingerprint is:
d9:5e:b9:22:b0:1c:c1:e5:f7:28:eb:a8:bb:f1:6c:65 t-pc2@t-pc2
The key's randomart image is:
+--[ RSA 2048 ]-----+
|      .                |
|    . 0                |
|   o . .               |
|  . + 0 .              |
| o S o +               |
| . +E+ . .             |
| . 000 o .             |
|+.o . .                |
|+++.                  |
+-----+
t-pc2@t-pc2:~/Downloads/charm-6.2-multicore-linux64/examples/charm++/hello/ldarray$
```

Joonis 6. Väljalõige ssh võtme genereerimisest.

12. Järgmiseks minna juurkataloogi ja sealt edasi .ssh kausta:

cd ~

cd .ssh/

### 13. Tekitada võtmed ja need lisada:

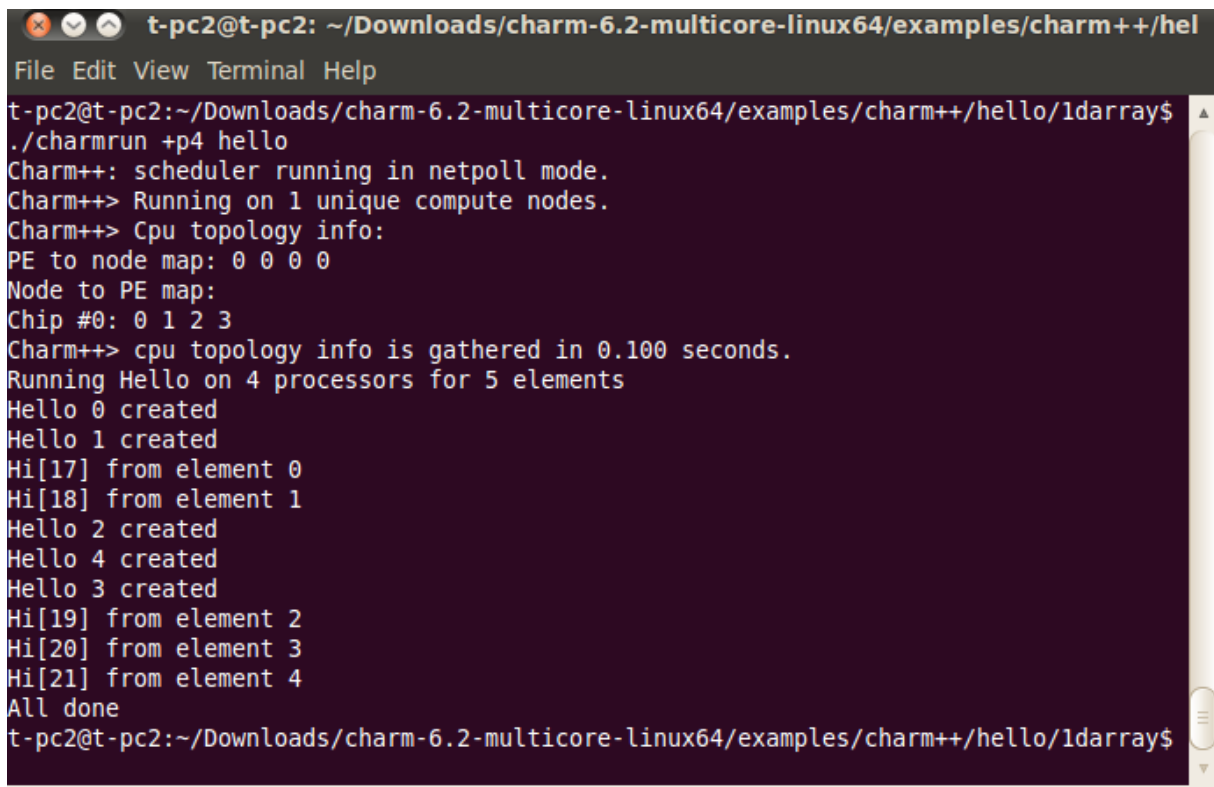
```
cat id_rsa.pub >> authorized_keys
```

```
ssh-add
```

**Vastus:** Identity added: /home/t-pc/.ssh/id\_rsa (/home/t-pc/.ssh/id\_rsa)

### 14. Nüüd võib koodi jooksutada:

```
./charmrun +p4 hello
```



```
t-pc2@t-pc2: ~/Downloads/charm-6.2-multicore-linux64/examples/charm++/hel
File Edit View Terminal Help
t-pc2@t-pc2:~/Downloads/charm-6.2-multicore-linux64/examples/charm++/hello/ldarray$
./charmrun +p4 hello
Charm++: scheduler running in netpoll mode.
Charm++> Running on 1 unique compute nodes.
Charm++> Cpu topology info:
PE to node map: 0 0 0 0
Node to PE map:
Chip #0: 0 1 2 3
Charm++> cpu topology info is gathered in 0.100 seconds.
Running Hello on 4 processors for 5 elements
Hello 0 created
Hello 1 created
Hi[17] from element 0
Hi[18] from element 1
Hello 2 created
Hello 4 created
Hello 3 created
Hi[19] from element 2
Hi[20] from element 3
Hi[21] from element 4
All done
t-pc2@t-pc2:~/Downloads/charm-6.2-multicore-linux64/examples/charm++/hello/ldarray$
```

Joonis 7. Väljalõige töötavast Charm++ programmist 4 protsessi peal.

Nüüd oleks õige aeg proovida CD plaadil olevat Charm++ programmi käivitada (failid asuvad „Charm++ maatriksite korrutamise“ kataloogis). Selles kataloogis on fail nimega „Makefile“ ja selles failis peab muutma esimest rida vastavalt oma Charm++ paketi asukohale. Testitavas arvutis oli selleks näiteks „/home/t-pc/Downloads“ kus oli lahti pakitud kaust „charm-6.2-multicore-linux64“. Seega ülemine rida näeb välja selline:

```
CHARMDIR = /home/t-pc/Downloads/charm-6.2-multicore-linux64
```

Pärast „Makefile“ muutmist saab sisestada lihtsalt käsu:

```
make
```

```
t-pc2@t-pc2: ~/Downloads/Lisad/Charm++ maatriksite korrutamine
File Edit View Terminal Help
t-pc2@t-pc2:~/Downloads/Lisad/Charm++ maatriksite korrutamine$ make
/home/t-pc2/Downloads/charm-6.2-multicore-linux64/bin/charmc main.ci
/home/t-pc2/Downloads/charm-6.2-multicore-linux64/bin/charmc multiply.ci
/home/t-pc2/Downloads/charm-6.2-multicore-linux64/bin/charmc -o main.o main.C
/home/t-pc2/Downloads/charm-6.2-multicore-linux64/bin/charmc -o multiply.o multiply.C
/home/t-pc2/Downloads/charm-6.2-multicore-linux64/bin/charmc -language charm++ -o multiply
main.o multiply.o
t-pc2@t-pc2:~/Downloads/Lisad/Charm++ maatriksite korrutamine$
```

Joonis 8. Charm++ programmi kompileerimine.

Nüüd võiks näite programmi käivitada käsuga:

```
./charmrun +p4 ./multiply 4
```

```
t-pc2@t-pc2: ~/Downloads/Lisad/Charm++ maatriksite korrutamine
File Edit View Terminal Help
t-pc2@t-pc2:~/Downloads/Lisad/Charm++ maatriksite korrutamine$ ./charmrun +p4 ./multiply 4
Charm++: scheduler running in netpoll mode.
Charm++> Running on 1 unique compute nodes.
Charm++> Cpu topology info:
PE to node map: 0 0 0 0
Node to PE map:
Chip #0: 0 1 2 3
Charm++> cpu topology info is gathered in 0.122 seconds.
 0  1  2  3
 4  5  6  7
 8  9 10 11
12 13 14 15

  X

 0  1  2  3
 4  5  6  7
 8  9 10 11
12 13 14 15

Running with 4 elements using 4 processors.
Start calculation

Tulemus:
56 62 68 74
152 174 196 218
248 286 324 362
344 398 452 506
t-pc2@t-pc2:~/Downloads/Lisad/Charm++ maatriksite korrutamine$
```

Joonis 9. Väljalõige Charm++ raamistikus maatriksite korrutamisest.

Viimane number näitab kui suuri maatrikseid omavahel korrutakse. Seda numbrit võib suurendada vastavalt soovile, kuid see number peab jaguma protsessorite arvuga (näiteks võib vabalt jooksutada 9x9 maatriksit kolme protsessori peal).



## MPI standardi paigaldamine

1. Esmalt paigaldada vajalikud failid:

```
sudo apt-get install libopenmpi-dev
```

```
sudo apt-get install openmpi-bin
```

2. MPI peaks olema juba paigaldatud ja proovida kompileerimist:

```
mpicc programm.c -o programmi_nimi
```

3. Ja programmi käivitamiseks:

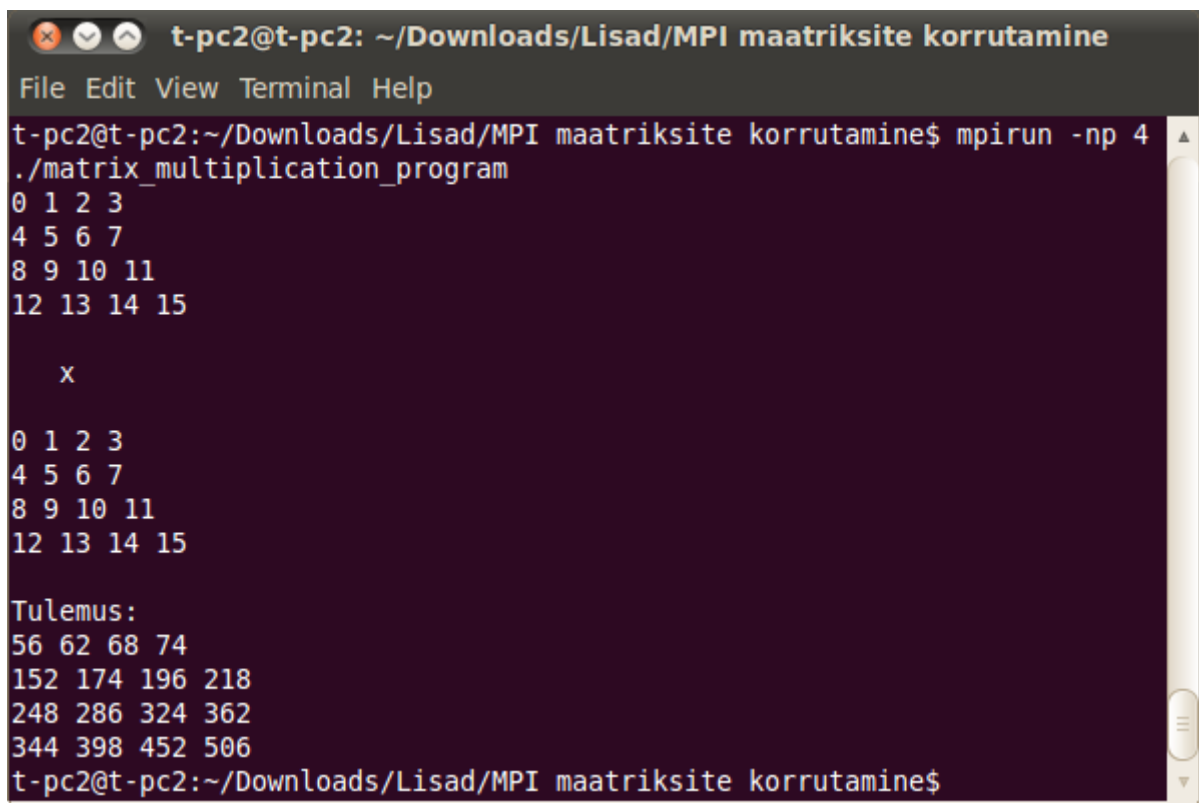
```
mpirun -n 4 ./programmi_nimi
```

Näiteks võib proovida CD plaadil oleva „matrix\_multiplication.c“ programmi kompileerida ja käivitada (fail asub „MPI maatriksite korrutamine“ kaustas). Lähtudes teisest punktist võib kompileerimine välja näha järgmine:

```
mpicc matrix_multiplication.c -o matrix_multiplication_program
```

Ja käivitada vastavalt loodud programmi:

```
mpirun -np 4 ./matrix_multiplication_program
```



```
t-pc2@t-pc2: ~/Downloads/Lisad/MPI maatriksite korrutamine
File Edit View Terminal Help
t-pc2@t-pc2:~/Downloads/Lisad/MPI maatriksite korrutamine$ mpirun -np 4
./matrix_multiplication_program
0 1 2 3
4 5 6 7
8 9 10 11
12 13 14 15

x

0 1 2 3
4 5 6 7
8 9 10 11
12 13 14 15

Tulemus:
56 62 68 74
152 174 196 218
248 286 324 362
344 398 452 506
t-pc2@t-pc2:~/Downloads/Lisad/MPI maatriksite korrutamine$
```

Joonis 10. Väljalõige maatriksite korrutamise programmi käivitamisest MPI standardis.