

Računalna astrofizika

Miljenko Čemeljić

Centar izvrsnosti za astronomiju, Varaždin

&

Silesian University, Institute of Physics in Opava,
Czech Republic

&

Nicolaus Copernicus Astronomical Center of the
Polish Academy of Sciences,
Warsaw, Poland

&

Academia Sinica Institute of Astronomy and
Astrophysics, Taipei, Taiwan, Visiting Scholar



SILESIA
UNIVERSITY
INSTITUTE OF PHYSICS
IN OPAVA



Sadržaj predavanja

- **1. dio:** Uloga računala u modernoj astronomiji i astrofizici, Linux, gnuplot, C
- **2. dio:** Instalacija, pokretanje i testiranje koda PLUTO, simulacija sferične eksplozije u 2D i 3D, instalacija Paraview i VisIt paketa i vizualizacija rezultata
- **3. dio:** Uvod u akrecijske diskove, simulacija diska, vizualizacija rezultata, računanje u paralelnom načinu rada i na clusterima i superkompjuterima.
- **4. dio:** Osnove Python-a, pisanje programa i njegovo izvršavanje, vizualizacija pomoću matplotlib biblioteke, pokretanje DUSTER koda za svemirsku prašinu
- **5. dio:** LaTeX, Overleaf, priprema izvješća

-provjerite web stranicu https://web.tiara.sinica.edu.tw/~miki/miki_RacunAF.html

Potrebni paketi: PLUTO source code, C-compiler (+MPI), Python (matplotlib, NumPy), Gnuplot, Paraview, VisIt, LaTeX.

Kratki pregled – 1. dio

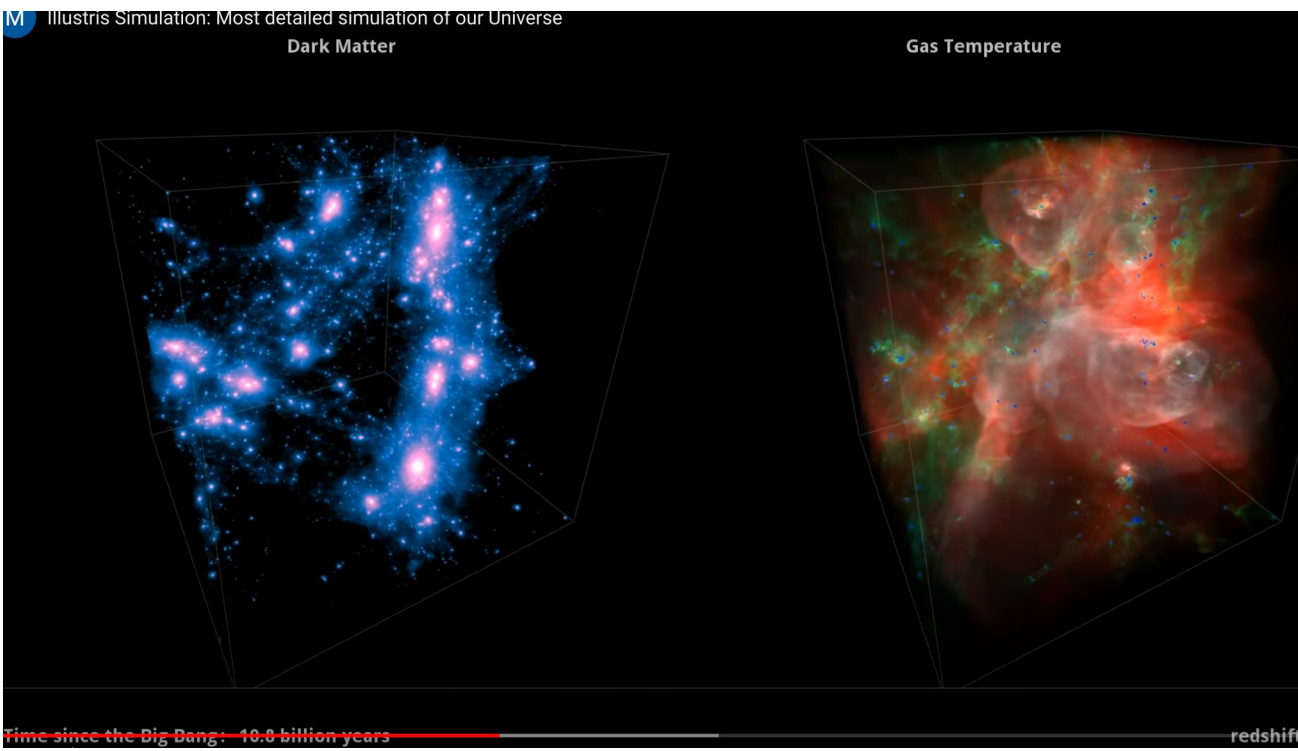
- Uloga računala u modernoj astrofizici
- “Simulacije” u usporedbi s “računanjem”.
- Linux shell
- Primjer instalacije paketa u Linuxu: gnuplot
- Osnove jezika C

Uloga računala u modernoj astrofizici-analitički modeli

- Promatranja u astronomiji su danas vrlo rijetko neposredna, obično je između promatrane pojave i njenog objašnjenja par koraka, koji uopće ne moraju biti “očiti”. Najbolji je primjer “promatranja” smeđih patuljaka, kad se traži na nebu tamne točke i onda pretpostavi da su to upravo smeđi patuljci koje se NE vidi. Tada se upotrijebi druge instrumente, koji bi vidjeli smeđe patuljke, na tim točkama i provjeri da li su stvarno tamo.
- Klasična Galileijeva promatranja pjega na Suncu i prstena Saturna isto uključuju veliku dozu pretpostavki i predznanja. Pokažite Saturn djeci i pitajte ih što vide!
- Danas npr. promatranja X i gama zraka uključuju vrlo apstraktne korake statističke obrade, koji uključuju vjerojatnost da nam iz nekog smjera dođe neki broj fotona-takve pretpostavke su često dosta proizvoljne i ovisne su o modelima.
- “Modeli” su ili vrlo aproksimativni, pojednostavljeni analitički računi, ili danas češće više ili manje pojednostavljene numeričke simulacije.
- Primjer pojednostavljenog analitičkog računa je račun strukture zvijezde-vrlo jednostavan, jer radi se o lopti plina. Dobije se dosta robustan opis koji dobro opisuje zvijezde općenito, u vrlo širokom području važenja zakona idealnog plina. To radi za površinu zvijezde, ali kad se upustimo u račune “pogonskog motora” zvijezde, ne radi, tamo nam treba znanje nuklearne fizike. Kad su probali objasniti energijski izvor zvijezde pomoću ugljena, ispalo je da bi svijetlila vrlo kratko (iz veličine zvijezde i njenog sjaja se lako da odrediti koliku energiju izrača zvijezda)-to je bio glavni problem astrofizike početkom XX stoljeća. Tek stotinjak godina zapravo znamo kako zvijezde “rade”!

Uloga računala u modernoj astrofizici-numeričke simulacije

- Simulacija stvaranja strukture u Svemiru, “Illustris”, koristi se za uspoređivanje stvaranja pojedinih sub-struktura s promatranim, kao i za modele stvaranja zvijezda i galaksija. Tu u biti nemamo druge niti vodilje osim simulacija.
- Najpoznatiji primjer danas je promatranje “sjene” crne rupe, to bi stvarno teško bilo nazvati “snimkom”, najbliže je, ali jako pojednostavljeno, reći “usrednjena snimka”.
- Promatranja gravitacijskih valova su idući stupanj apstrakcije, moramo znati što tražimo u signalu i onda to izvući iz šuma koji je mnogo redova veličine jači od signala! Tražimo signal promjene duljine reda veličine stomilijuninke promjera atoma, i to ne izolirani signal nego utopljen u mnogo bliže jake izvore.



Simulacije u usporedbi s računanjem

- Račun je zapravo evaluacija: ubacite brojeve u zadani algoritam i dobijete rezultat. Za to morate znati analitički izraz. Možete koristiti kalkulator ili računalo za dobivanje brojeva i crtanje npr. putanje metka, ali jednadžbu za dobiti rješenje, ili približni (ili točan) računski postupak, znate unaprijed.
- U slučaju simulacije, ne znate jednadžbu za riješiti problem. Ili čak znate da ju je nemoguće dobiti (takav slučaj je magnetizirani disk koji ćemo simulirati tokom ovih predavanja). Postavite jednadžbe, npr. diferencijalne jednadžbe i koristite numeričku metodu za pronalaženje rješenja. To se obično radi u vremenskim koracima, a rješenje dobivamo do određene preciznosti. Ne znamo analitički izraz za rješenje, imamo samo brojeve= numeričku simulaciju.
- Zašto PLUTO? Koristim i koristio sam i druge kodove, ali PLUTO se stalno razvija (i upotrebljavan je u raznim problemima) i razvoj se slijedi u priručniku i testnim primjerima koje dobijete uz kod- što nije tako često slučaj u svijetu kodiranja, jer ljudi nemaju vremena za njihovo pisanje. PLUTO pruža priliku korisnicima da prođu kroz još uvijek strmu krivulju učenja u najkraćem mogućem roku. Provjerite i Doxygen html datoteke u PLUTO/ Doc .

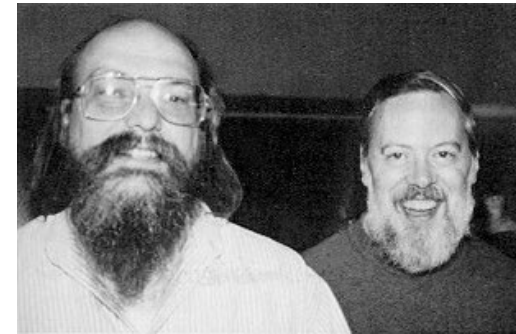


Linux

Tux, službena grafička maskota Linuxa.



- Bio jednom Linux Torvald. Wrong: ispravno je Linus Benedict Torvalds. Rođen 1969, naravno.
- Bio je jedan od programera koji su napisali linux kernel (1991), u biti radilo se o modifikaciji Unixa (zapamtite si UNIX kao “**sve je datoteka**”), operativnog sistema koji se razvio iz originalnog AT&T Unixa, čiji razvoj je započeo (naravno) 1969. u Bell Labs istraživačkom centru (Ken Thompson, Dennis Ritchie i dr.)
- Linux je započeo finski student Linus Torvalds kao vlastiti projekt, koji se od malog broja datoteka (bash, gcc) od 1991 razvio do 23.3 miliona linija (bez komentara), koji je pod GNU (“GNU’s not Unix”) General Public License v2, dakle po pravilima Open Source inicijative. On je smislio i git. Ideologija: geek hipiji na djelu, pogledajte Richarda Stallmana, glavnog i odgovornog za GNU! (St.IGNUcius). On je smislio Copyleft: [a method of making intellectual property reusable and modifiable without any restrictions, except that anything new produced using the original asset must also be available freely. This can apply to everything from works of art to software.], koji je osnova GNU.



Linux kernel, shell, X, ...

- Kernel je osnovni operacijski sistem za manipulaciju datotekama, diskovima, umrežavanje itd
- Shell je program preko kojeg komunicirate s računalom preko konzole-terminala. GUI=graphical user interface radi isto, samo sa slikicama. Osnovne komande su ls, cd, cp, mv, rm, mkdir, grep. Editor vi (vi vi vi), nano, ... Koristite ćemo BASH=Bourne again shell, koji je često default. Ostali imaju slične osnovne funkcije.
- X je grafički sistem koji omogućava korištenje prozora, menija, ikona, miša i ostalog u GUI. KDE, GNOME, Unity i slično su samo kompleksnije strukture izgrađene pomoću X.
- Danas ćete obično naići na dvije distribucije Linuxa: Red Hat i Debian, iako se često naiđe i na nezavisne, kao npr. Arch, kod kojih se morate više potruditi za instalaciju svega što vam treba. Android je isto Linux kernel koji je usko isprepleten sa Google aplikacijama.
- Na svakom sistemu pod linuxom obično možete imati mnogo korisnika (user) ali jedan je superuser, tzv. "root", koji ima dostup do svega. Obično je to administrator. Ubuntu je maknuo "root" u pozadinu davanjem prava administracije korištenjem "sudo" komande (super user DO), tako da se npr. Može instalirati neki novi program bez toga da se bude administratorom cijelog sistema (superuser).

Linux shell

Osnovne komande u jeziku shell su ls, cd, cp, mv, rm, mkdir, grep. Editor vi (vi vi vi), nano, ... Ako se želite poigrati, u shell možete napraviti vrlo ozbiljne programe.

```
Terminal - miki@mikic: ~/Desktop/VzAstroCentar/PlutoAstrocPredavanja
File Edit View Terminal Tabs Help
miki@mikic:~/Desktop/VzAstroCentar/PlutoAstrocPredavanja$ pwd
/home/miki/Desktop/VzAstroCentar/PlutoAstrocPredavanja
miki@mikic:~/Desktop/VzAstroCentar/PlutoAstrocPredavanja$ ls
330px-Linus_Torvalds.jpeg          mikiRacAfLeCsVz1.pdf
ArhivneRacAF                       mikiRacAfLeCsVz.odp
cemeljic_Simuls2022.txt           stallmanIGNU.png
cemeljic_Simuls2022.txt~         stallman.png
Ken_Thompson_and_Dennis_Ritchie--1973.jpg Tux.png
mikiRacAfLeCsVz1.odp
```

```
miki@mikic:~/Desktop/VzAstroCentar/PlutoAstrocPredavanja$ ls -lh
total 24M
-rw-rw-r-- 1 miki miki 44K lis 7 14:03
drwxrwxr-x 2 miki miki 4,0K paž 29 12:23
-rw-rw-r-- 1 miki miki 2,4K paž 29 01:26
-rw-rw-r-- 1 miki miki 2,2K paž 21 14:46
-rw-rw-r-- 1 miki miki 20K lis 7 13:51
pg
-rw-rw-r-- 1 miki miki 671K lis 1 11:09
-rw-rw-r-- 1 miki miki 214K lis 1 11:09
-rw-r--r-- 1 miki miki 22M lis 7 16:06
-rw-rw-r-- 1 miki miki 185K lis 7 15:59
-rw-rw-r-- 1 miki miki 1,7M lis 7 15:36
-rw-rw-r-- 1 miki miki 12K lis 7 10:56
miki@mikic:~/Desktop/VzAstroCentar/Pluto
```

```
Terminal - miki@mikic: ~/Desktop/VzAstroCentar/PlutoAstrocPredavanja
File Edit View Terminal Tabs Help
GNU nano 4.8 cemeljic_Simuls2022.txt
MC211022--program predavanja za Astro Centar Vž:
Numeričke simulacije u astronomiji
Motivacija: Uvod u numeričke simulacije s primjenom u astronomiji.
Ova kratka radionica obuhvaća, u 10 školskih sati (5 blokova od po
2 sata predavanja sa vježbama), osnove potrebne za sprovođenje
numeričkih simulacija, s primjerima iz fizike plazme i astrofizike.
Usput ćemo proći kroz osnove jezika shell, C i Python, paketa za
vizualizaciju Paraview i paketa LaTeX za obradu teksta. Cilj je
upoznati polaznike sa teoretskim osnovama, i odmah ih praktično
upotrijebiti. Predavanja će biti izvedena u "hands-on" načinu: uz
teorijski dio, svaki polaznik ima pred sobom računalo (po želji:
svoje osobno ili osigurano od strane Astro Centra) i slijedi upute
predavača u provođenju vježbi.

Program:
Predavanje 1:
-Kratki uvod u numeričke simulacije sa naglaskom na njihovu ulogu
u astrofizici danas
-Uvod u korištenje Linux sistema Ubuntu, praktične vježbe upotrebe
[ Read 52 Lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^I To Spell ^_ Go To Line
```

Instalacija paketa u linuxu

Trebat će nam kasnije GNUPLOT, program za crtanje, pa ako ga već nemate instaliranog, treba ga instalirati. Kako se to radi?

Način je sličan na svim linuxima, iako se same komande mogu razlikovati (sudo na Ubuntu za potvrdu kao superuser, da se može instalirati nove programe bez da se bude "root"):

sudo apt-get install (aplikacija), npr.

- `miki@mikic:$ sudo apt-get install gnuplot`
- Kad instaliramo, pokrećemo sa
`miki@mikic:$ gnuplot`

Pokretanje aplikacije u linuxu

```
x - +
Terminal - miki@mikic: ~/Gnuplot
File Edit View Terminal Tabs Help
miki@mikic:~/Gnuplot$ gnuplot

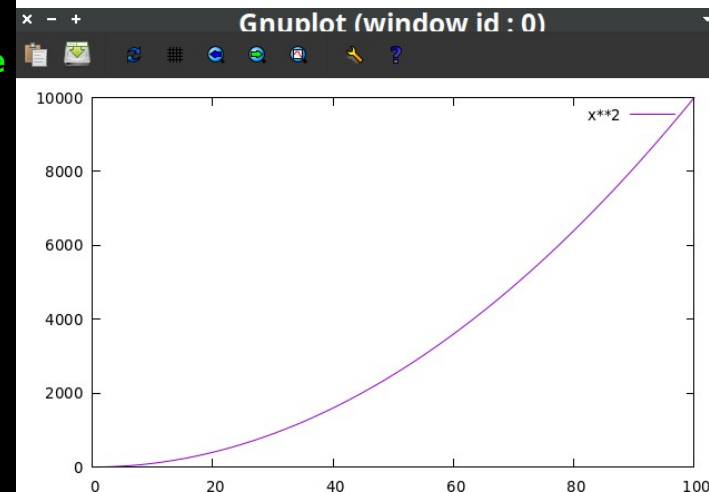
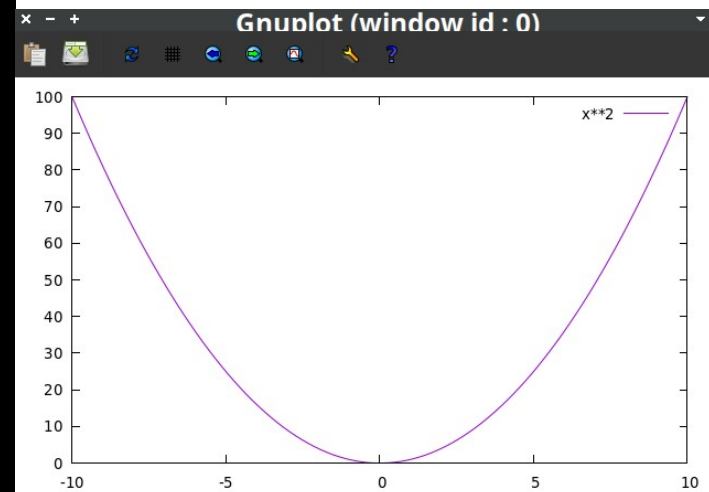
G N U P L O T
Version 5.2 patchlevel 8    last modified 2019-12-01

Copyright (C) 1986-1993, 1998, 2004, 2007-2019
Thomas Williams, Colin Kelley and many others

gnuplot home:      http://www.gnuplot.info
faq, bugs, etc:   type "help FAQ"
immediate help:   type "help" (plot window: hit 'h')

Terminal type is now 'wxt'
gnuplot> set title "Plot"
gnuplot> set title font ",20" norotate
gnuplot> help sum
A summation expression has the form
    sum [<var> = <start> : <end>] <expression>
<var> is treated as an integer variable that takes on successive integral
values from <start> to <end>. For each of these, the current value of
<expression> is added to a running total whose final value becomes the value
of the summation expression.
Examples:
    print sum [i=1:10] i
           55.
    # Equivalent to plot 'data' using 1:($2+$3+$4+$5+$6+...)
    plot 'data' using 1 : (sum [col=2:MAXCOL] column(col))
It is not necessary that <expression> contain the variable <var>.
Although <start> and <end> can be specified as variables or expressions,
their value cannot be changed dynamically as a side-effect of carrying
out the summation. If <end> is less than <start> then the value of the
summation is zero.

gnuplot> print 2+2
4
gnuplot> plot x**2
gnuplot> plot [0:100] x**2
gnuplot> plot x**3
gnuplot> exit
```



Osnove jezika C

- C je vrlo mali programski jezik. Ono što ga čini moćnim su biblioteke koje dodajemo za razne upotrebe.
- “Compiler” je program koji prevodi naše komande u C
- U Linuxu, obično je by default instaliran gcc, gnu C compiler
- Obično se, kad se uči novi jezik, nauči pisati “hello, world”. Dakle, kako unijeti taj tekst i ispisati ga u terminalu?
- Ako ne znamo komandu, često postoji manual u sistemu, ili help ili tako nešto. Ovdje znamo da želimo upotrijebiti compiler gcc, pišemo
man gcc
I dobijemo upute: gcc -o program.out program.c
gdje je program ime našeg programa.
- Programiranje se uči na primjerima, evo hello world programa hello.c:
include <stdio.h>
main()
{
printf("hello, world\n");
}
- stdio.h je za “standard input output” , u našem slučaju default je ekran terminala.
- Program ima glavni dio, unutar kojeg su komande.
- Slijedimo upute od prije: pišemo
gcc -o hello.out hello.c
- Ne obaziremo se na warning, to se obično nekom opcijom izbjegava, ali ako ne dobijemo ERROR, vrijedi probati pokrenuti:
./hello.out
- Dobijemo ispisano, neiznenađujuće:
hello, world
- Napisali ste svoj prvi program u c, kompajlirali ga i pokrenuli! Ostalo je učenje na primjerima i vaša kreativnost!
- Iduća dva primjera skinete sa linka na dnu stranice https://web.tiara.sinica.edu.tw/~miki/miki_RacunAF.html

Primjer programa u C: Fahrtheiti u Celsiuse, fahrrels.c

```
#include <stdio.h>

/* prints Fahr-Cels table for fahr=0,20,...300 */
main()
{
float fahr, celsius;
int lower, upper, step;

lower = 0; /*lower limit*/
upper = 300; /*upper limit*/
step = 20; /*step size*/

printf("\n", fahr, celsius);
printf("-----\n", fahr, celsius);
printf("Exercise: heading above table\n", fahr, celsius);
printf("-----\n", fahr, celsius);

fahr = lower;
while (fahr <= upper) {
celsius = 5.0 * (fahr-32.0) / 9.0 ;
printf("%3.0f %6.1f\n", fahr, celsius);
fahr = fahr + step;
}
}
```

Primjer programa u C: Fahrenheiti u Celsiuse

- Kompilacija i pokretanje kao prije:

```
gcc -o fahrcels.out fahrcels.c
```

```
./fahrcels.out
```

- Dobijemo ispis tabele:

```
miki@mikic:~/CMiki$ fahrcels.out
```

```
-----
```

```
Exercise: heading above table
```

```
-----
```

```
0 -17.8  
20 -6.7  
40 4.4  
60 15.6  
80 26.7  
100 37.8  
120 48.9  
140 60.0  
160 71.1  
180 82.2  
200 93.3  
220 104.4  
240 115.6  
260 126.7  
280 137.8  
300 148.9
```

```
miki@mikic:~/CMiki$
```

Primjer kompaktnijeg zapisa programa: f2ahrcels.c

```
#include <stdio.h>
#define LOWER 0
#define UPPER 300
#define STEP 20
/* prints Fahr-Cels table for fahr=0,20,...300 */
main()
{
    int fahr;

    for (fahr = UPPER; fahr >= LOWER; fahr = fahr - STEP)
        printf("%3d %6.1f\n", fahr, 5.0 * (fahr-32.0) / 9.0);
}
```

- Kompilacija i pokretanje kao prije:

```
gcc -o f2ahrcels.out f2ahrcels.c
```

```
./f2ahrcels.out
```

```
miki@mikic:~/CMiki$ f2ahrcels.out
```

```
300 148.9
```

```
280 137.8
```

```
260 126.7
```

```
240 115.6
```

```
220 104.4
```

```
....
```

```
20 -6.7
```

```
0 -17.8
```

Primjer kompaktnijeg zapisa programa: f2ahrcels.c

- Probajmo dodati natpis iznad tabele kao prije, samo skopirajmo:

```
#include <stdio.h>
#define LOWER 0
#define UPPER 300
#define STEP 20
/* prints Fahr-Cels table for fahr=0,20,...300 */
main()
{
int fahr;
printf("\n", fahr, celsius);
printf("-----\n", fahr, celsius);
printf("Exercise: heading above table\n", fahr, celsius);
printf("-----\n", fahr, celsius);
for (fahr = UPPER; fahr >= LOWER; fahr = fahr - STEP)
    printf("%3d %6.1f\n", fahr, 5.0 * (fahr-32.0) / 9.0);
}
```

- Kad skompajliramo dobijemo error:

```
miki@miki:~/CMiki$ gcc -o f2ahrcels.out f2ahrcels.c
```

- f2ahrcels.c:6:1: warning: return type defaults to 'int' [-Wimplicit-int]

```
6 | main()
```

```
| ^~~~
```

- f2ahrcels.c: In function 'main':

- f2ahrcels.c:10:20: error: 'celsius' undeclared (first use in this function)

```
10 | printf("\n", fahr, celsius);
```

```
|           ^~~~~~
```

- f2ahrcels.c:10:20: note: each undeclared identifier is reported only once for each function it appears in

Primjer kompaktnijeg zapisa programa: f2ahrcels.c

- Očito fali deklaracija "celsius". Sve varijable treba deklarirati!
- Dodamo: int fahr, celsius;
- Skompajliramo, ide!, dobijemo ispis tabele sa natpisom iznad.
- Program za brojenje karaktera u unosu, charcount1.c:

```
include <stdio.h>
/* count chars in input but have to press control+D to exit */
main()
{
    long nc;
    nc = -1;
    while (getchar() != EOF)
        ++nc;
    printf("%ld\n", nc);
}
```

- NAPRAVI: Probajmo napraviti bez potrebe za ctrl+D, da nakon enter ispiše izađe: charcount2.c:

```
include <stdio.h>
/* count chars in input */
main()
{
    double nc;
    for(nc = 0; getchar() != EOF ; ++nc)
        ;
    printf("%.0f\n", nc-1, EOF);
}
```

- gcc -o charcount2.out charcount2.c