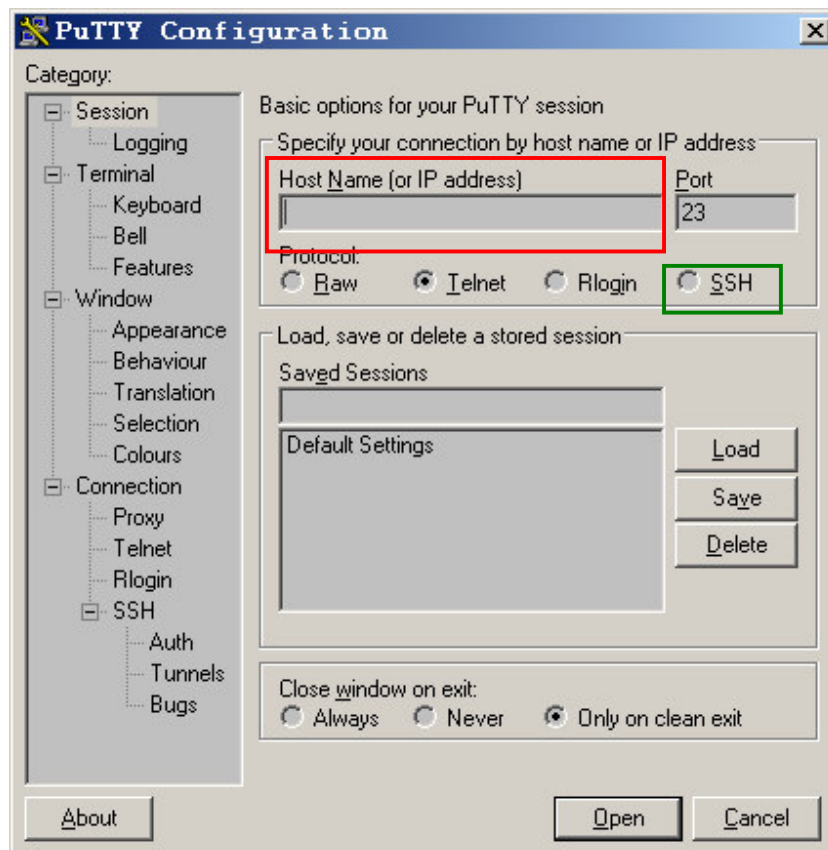


Programmieren am Pluto

In dieser Dokumentation bedeuten GROSSGESCHRIEBNE Worte Tasten auf der Tastatur, und [klammern] das anstelle ein freier Text eingegeben werden kann.

1.0 Verbinden zum Pluto

Dazu wird das Programm putty unter Windows gestartet (alternativ von <http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe> heruntergeladen). Es kann auch jeder anderer beliebige SSH Client verwendet werden; die Befehle auf der Konsole bleiben die selben (ab Kapitle 2).



Hier den Servernamen eingeben:

pluto.tugraz.at

Dieses Protokoll auswählen – SSH


```
altihar@pluto:~/esp
pluto.tugraz.at

[altihar@pluto ~]$ ls -l
total 152
-rwxr-xr-x 1 altihar unixuser 4826 Jun 27 23:51 a.out
drwx----- 2 altihar unixuser 83 Oct 9 2006 Desktop
drwxr-xr-x 2 altihar unixuser 4096 Oct 5 00:12 esp
drwxr-xr-x 3 altihar unixuser 42 May 6 23:15 etech
-rw-r--r-- 1 altihar unixuser 14641 May 10 16:53 log2.txt
-rw-r--r-- 1 altihar unixuser 74633 May 10 21:57 log.txt
lrwxrwxrwx 1 altihar unixuser 22 Sep 26 2006 Novell-L -> /mnt/ncpfs/a/altiha
lrwxrwxrwx 1 altihar unixuser 22 Sep 26 2006 Novell-M -> /mnt/ncpfs/a/altiha
-rwxr-xr-x 5 altihar unixuser 4096 Jun 14 14:23 sep
-rw-r--r-- 1 altihar unixuser 207 Jun 27 22:51 test.c
[altihar@pluto ~]$ cd esp
[altihar@pluto esp]$ ls -l
total 68
-rw-r--r-- 1 altihar unixuser 14 Oct 4 23:29 helloWorld.c
-rw-r--r-- 1 altihar unixuser 14 Oct 4 23:29 helloWorld.o
-rwxr-xr-x 1 altihar unixuser 4683 Oct 4 23:29 helloWorld.sh
-rw-r--r-- 1 altihar unixuser 268 Oct 4 23:29 helloWorld.c
-rwxr-xr-x 1 altihar unixuser 142 Oct 4 23:54 test
-rwxr-xr-x 1 altihar unixuser 121 Oct 5 00:09 test2.sh
-rwxr-xr-x 1 altihar unixuser 64 Oct 5 00:11 test3.sh
-rw-r--r-- 1 altihar unixuser 15 Oct 5 00:00 test.txt
[altihar@pluto esp]$
```

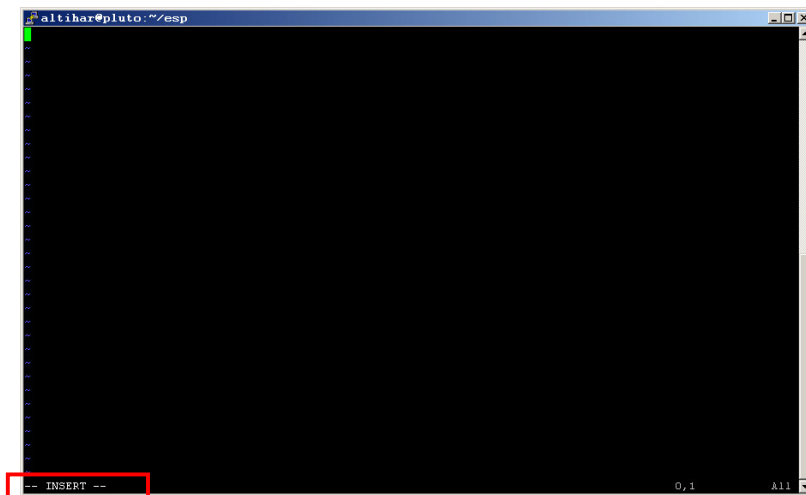
Zeigt den aktuellen Ordnerinhalt an; hier den des Benutzers

Wechselt in den Ordner ESP

Zeigt an das man mit dem Benutzer altihar auf dem Rechner Plut eingeloggt ist und sich im Verzeichnis esp befindet

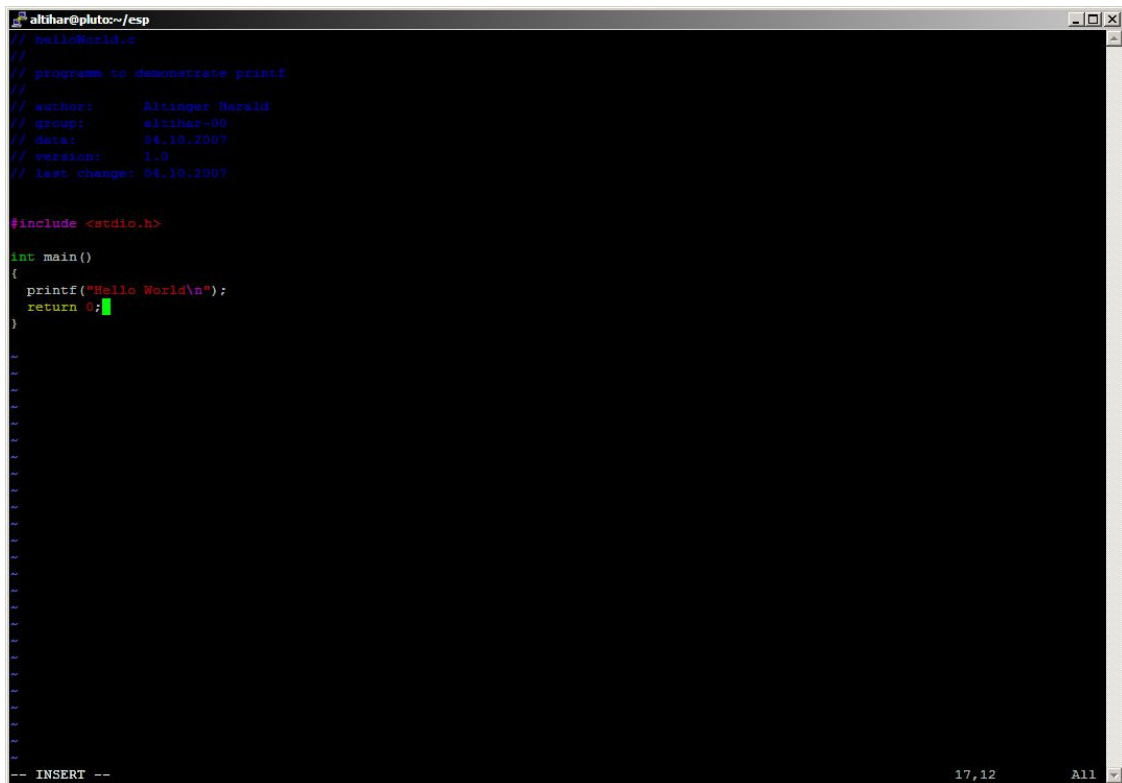
Mit dem Befehl `cd [verzeichnisname]` kann man in das erstellte Verzeichnis wechseln

Mit dem Befehl `cd ..` kann man in das vorherige Verzeichnis zurückwechseln.



```
altihar@pluto:~/esp
-- INSERT --
```

Nun kann Text eingegeben werden; VI ist in der Lage C Befehle zu erkennen und stellt die Syntax in unterschiedlichen Farben dar:



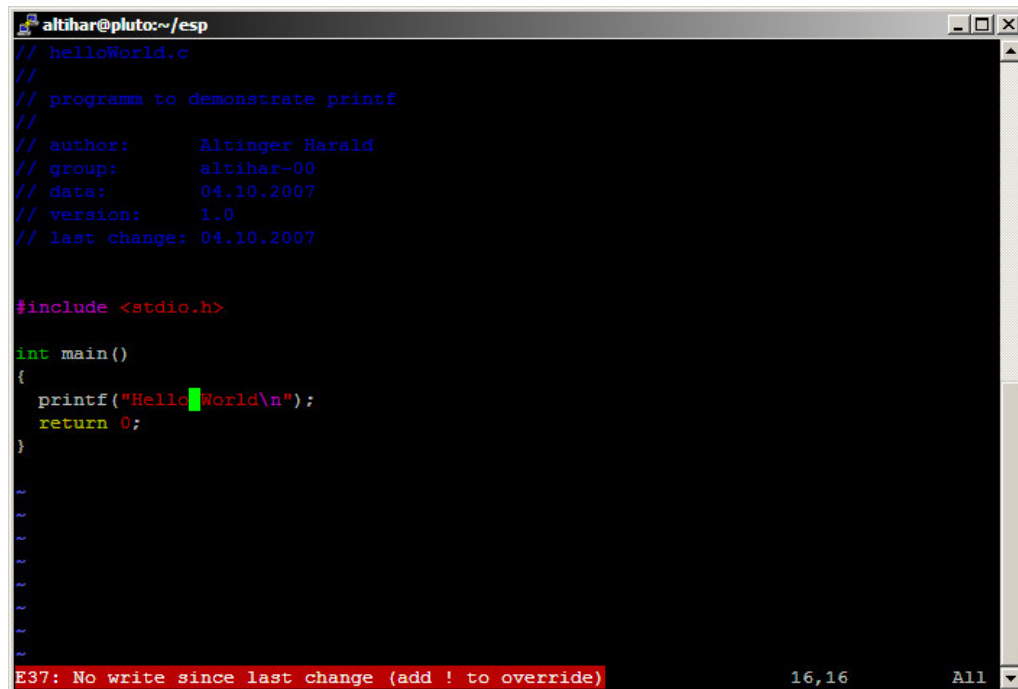
```
altihar@pluto:~/esp
// helloWorld.c
//
// program to demonstrate printf
//
// author:    Altinger Harald
// group:    altihar-08
// date:     04.10.2007
// version:  1.0
// last change: 04.10.2007

#include <stdio.h>

int main()
{
    printf("Hello World\n");
    return 0;
}

-- INSERT --
```


Hat man eine Aenderung nicht gespeichert, erscheint folgende Fehlermeldung:



```
altihar@pluto:~/esp
// helloWorld.c
//
// programm to demonstrate printf
//
// author:      Altinger Harald
// group:      altihar-00
// data:       04.10.2007
// version:    1.0
// last change: 04.10.2007

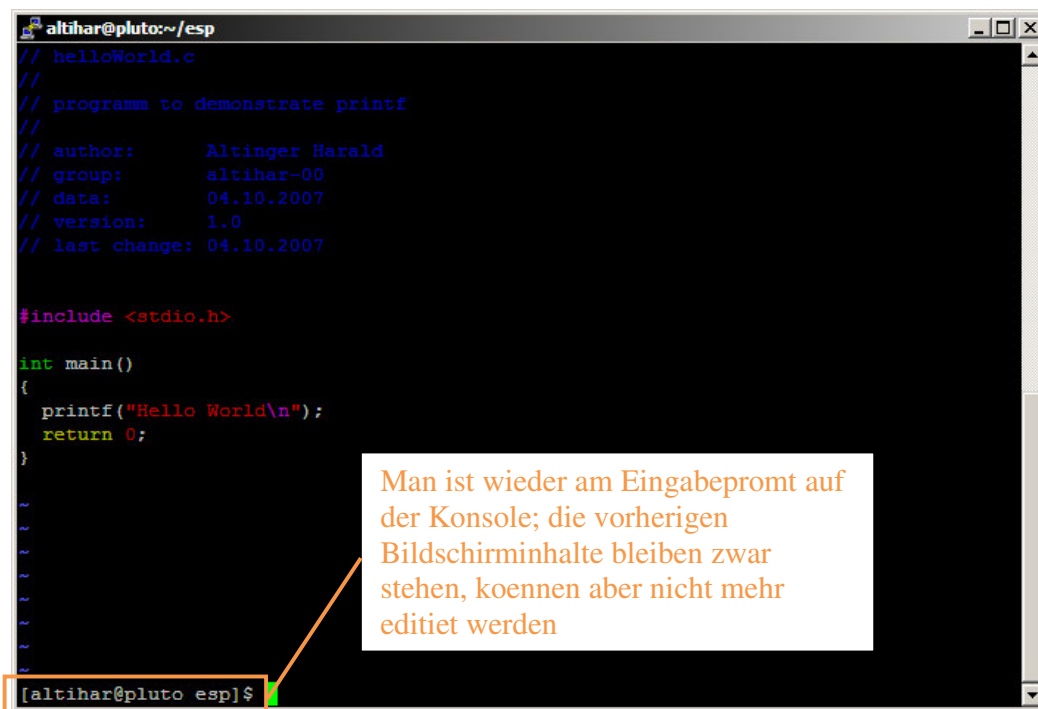
#include <stdio.h>

int main()
{
    printf("Hello World\n");
    return 0;
}

~
~
~
~
~
~
E37: No write since last change (add ! to override) 16,16 All
```

Dazu einfach noch einmal ESC druecken und den Befehl :q! eingeben, damit gehen alle Aenderungen verloren; oder durch den Befehl :w doch speichern.

Nach dem Befehl :q ist man wieder in der Unix kommandozeile:



```
altihar@pluto:~/esp
// helloWorld.c
//
// programm to demonstrate printf
//
// author:      Altinger Harald
// group:      altihar-00
// data:       04.10.2007
// version:    1.0
// last change: 04.10.2007

#include <stdio.h>

int main()
{
    printf("Hello World\n");
    return 0;
}

~
~
~
~
~
~
[altihar@pluto esp]$
```

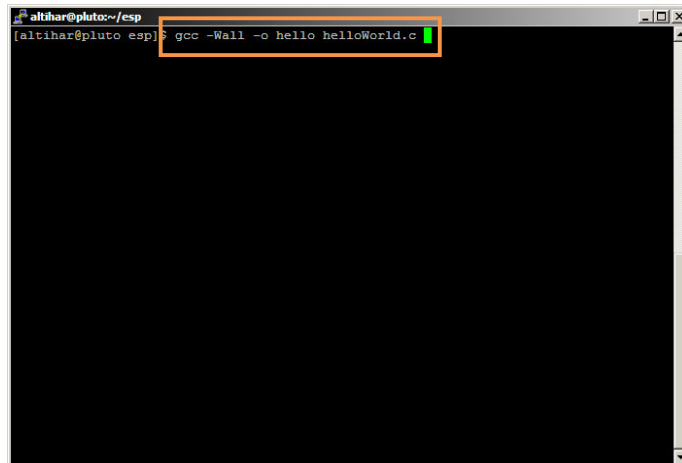
Man ist wieder am Eingabeprompt auf der Konsole; die vorherigen Bildschirmhalte bleiben zwar stehen, koennen aber nicht mehr editiert werden

4.0 Der Compiler

Nun muss das Programm kompiliert werden; dazu den Befehl:

```
gcc -Wall -o [programmname - kann frei gewählt werden] [Dateiname - ist das zuvor mit vi erstellte *.c File]
```

eingeben.



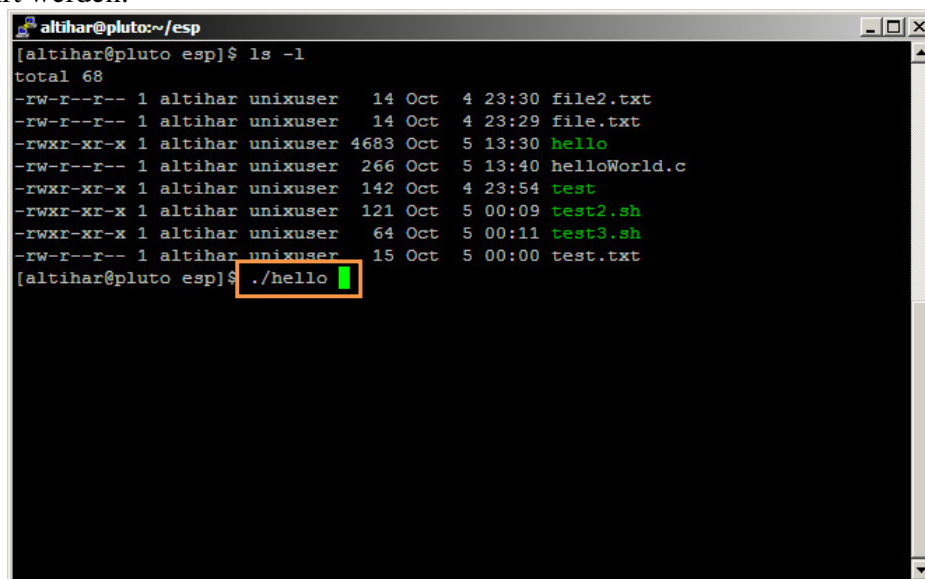
```
altihar@pluto:~/esp
[altihar@pluto esp] gcc -Wall -o hello helloWorld.c
```

Nun wurde ein ausführbares Programm mit dem angegebenen Programmnamen erstellt, hier hello.

Dies kann durch die eingabe von

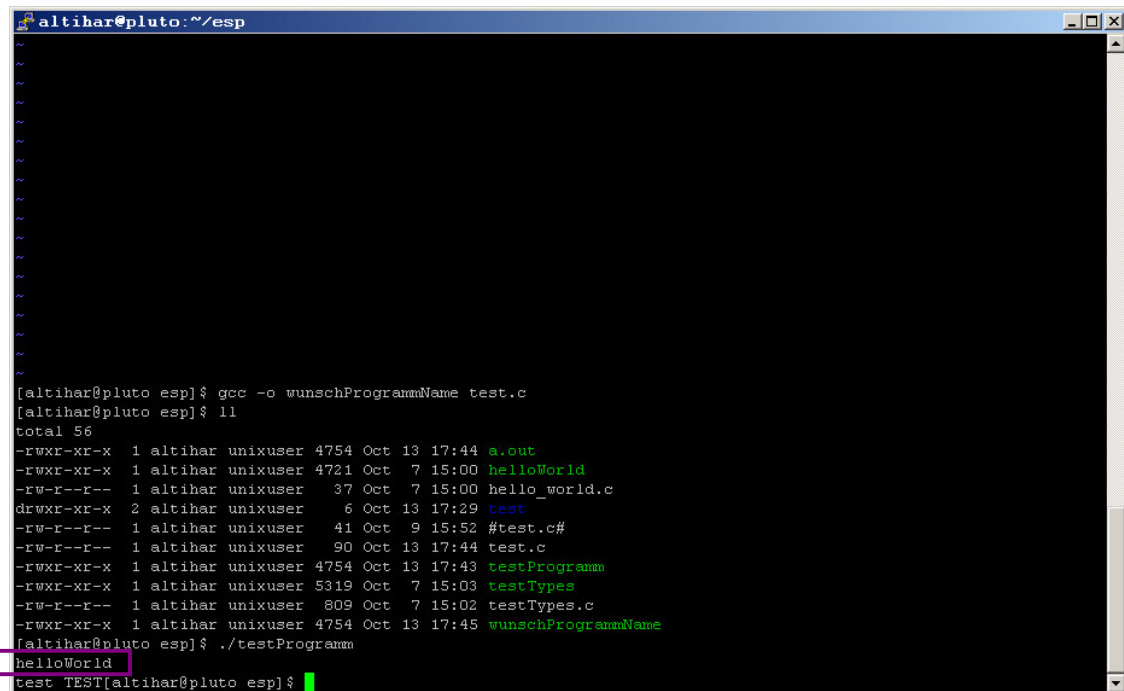
```
./[Programmname]
```

Ausgeführt werden.



```
altihar@pluto:~/esp
[altihar@pluto esp]$ ls -l
total 68
-rw-r--r-- 1 altihar unixuser  14 Oct  4 23:30 file2.txt
-rw-r--r-- 1 altihar unixuser  14 Oct  4 23:29 file.txt
-rwxr-xr-x 1 altihar unixuser 4683 Oct  5 13:30 hello
-rw-r--r-- 1 altihar unixuser  266 Oct  5 13:40 helloWorld.c
-rwxr-xr-x 1 altihar unixuser  142 Oct  4 23:54 test
-rwxr-xr-x 1 altihar unixuser  121 Oct  5 00:09 test2.sh
-rwxr-xr-x 1 altihar unixuser   64 Oct  5 00:11 test3.sh
-rw-r--r-- 1 altihar unixuser   15 Oct  5 00:00 test.txt
[altihar@pluto esp]$ ./hello
```

Das Programm hat die programmierten Befehle abgearbeitet:



```
altihar@pluto:~/esp
[altihar@pluto esp]$ gcc -o wunschProgrammName test.c
[altihar@pluto esp]$ ll
total 56
-rwxr-xr-x 1 altihar unixuser 4754 Oct 13 17:44 a.out
-rwxr-xr-x 1 altihar unixuser 4721 Oct 7 15:00 helloWorld
-rw-r--r-- 1 altihar unixuser 37 Oct 7 15:00 hello_world.c
drwxr-xr-x 2 altihar unixuser 6 Oct 13 17:29 test
-rw-r--r-- 1 altihar unixuser 41 Oct 9 15:52 #test.c#
-rw-r--r-- 1 altihar unixuser 90 Oct 13 17:44 test.c
-rwxr-xr-x 1 altihar unixuser 4754 Oct 13 17:43 testProgramm
-rwxr-xr-x 1 altihar unixuser 5319 Oct 7 15:03 testTypes
-rw-r--r-- 1 altihar unixuser 809 Oct 7 15:02 testTypes.c
-rwxr-xr-x 1 altihar unixuser 4754 Oct 13 17:45 wunschProgrammName
[altihar@pluto esp]$ ./testProgramm
helloWorld
test TEST[altihar@pluto esp]$
```

Wenn man sich ausloggen will gibt man den befehl:

exit

ein.

Darauf hin wird die SSH Verbindung zum Pluto beendet und Putty geschlossen.

5.0 Appendix:

Wichtige UNIX befehle:

<http://www.urz.uni-heidelberg.de/UnixCluster/Hinweise/Hilfe/Kommando/help.shtml>

Putty fuer Windows:

<http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>

SSH Client fuer MacOS:

<http://www.rbrowser.com/>

SSH Client unter Linux:

eine Konsole offnen und eingeben:

```
ssh -l [Benutzername] [name oder adresse des rechners]
```

z.B.: mit dem Benutzer username auf den pluto verbinden:

```
ssh -l username pluto.tugraz.at
```