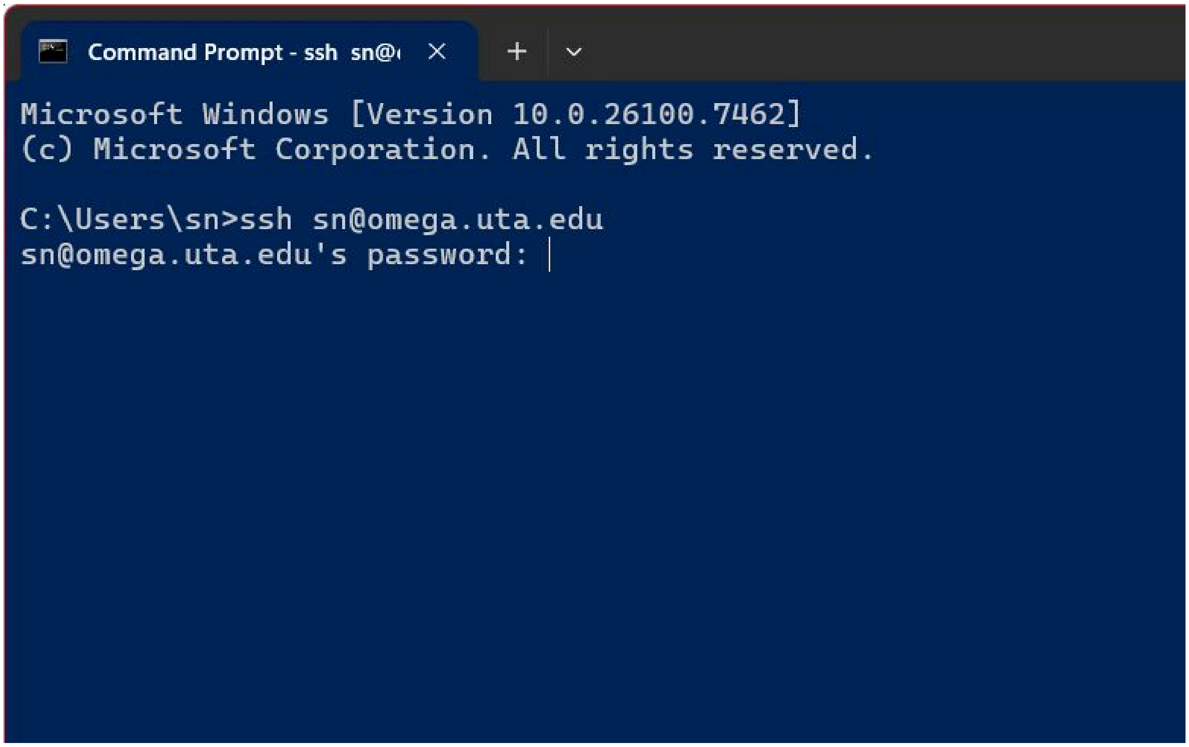
LAB #1 (01/12, 1/14, 1/16)

Logging in `omega`

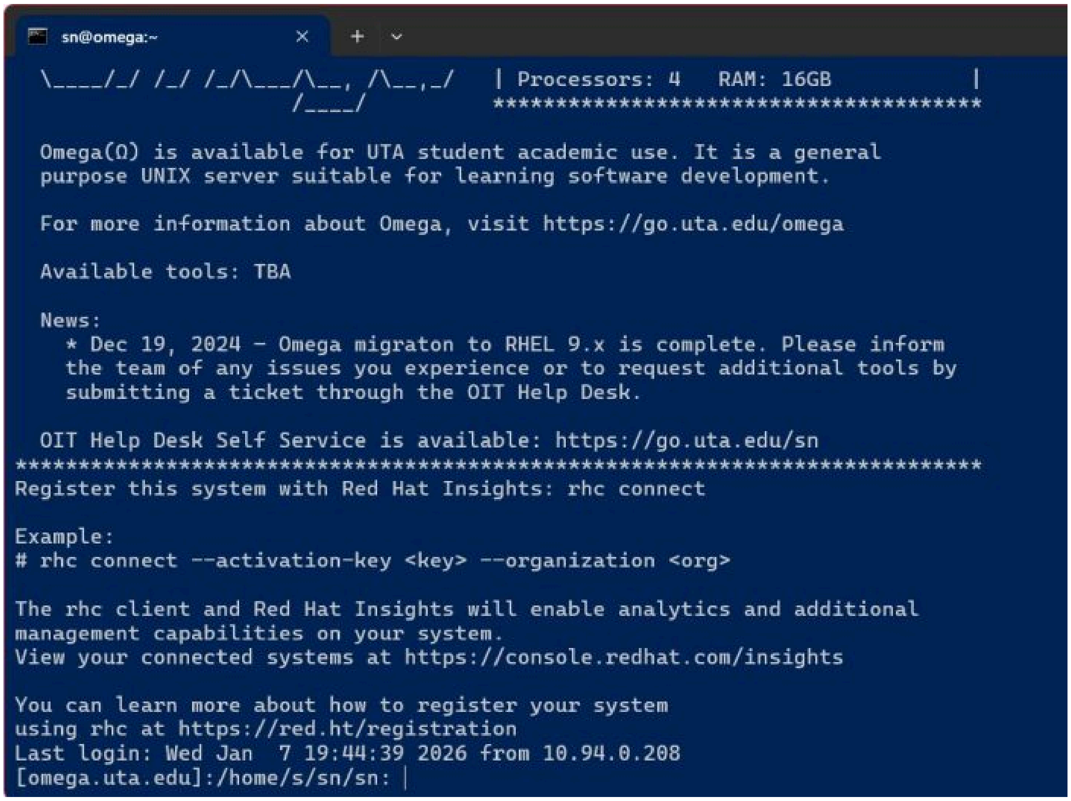
If you are on campus, you can login to **`omega.uta.edu`** from any computer on UTA wifi. If you are off campus, you must use Pulse Secure VPN first.

- Windows users:
  - Enter **cmd** in the search box and enter **ssh abc1234@omega.uta.edu** where **`abc1234`** is to be replaced by your NetID.
  - Enter **`ssh abc1234@omega.uta.edu`** in the search box



```
Microsoft Windows [Version 10.0.26100.7462]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sn>ssh sn@omega.uta.edu
sn@omega.uta.edu's password:
```



```
  \____/_/ /_/ /_/\___/\__, /\__,_/   | Processors: 4   RAM: 16GB          |
                      /____/          ***************************************

  Omega(Ω) is available for UTA student academic use. It is a general
  purpose UNIX server suitable for learning software development.

  For more information about Omega, visit https://go.uta.edu/omega

  Available tools: TBA

  News:
    * Dec 19, 2024 - Omega migraton to RHEL 9.x is complete. Please inform
    the team of any issues you experience or to request additional tools by
    submitting a ticket through the OIT Help Desk.

  OIT Help Desk Self Service is available: https://go.uta.edu/sn
*****************************************************************************
Register this system with Red Hat Insights: rhc connect

Example:
# rhc connect --activation-key <key> --organization <org>

The rhc client and Red Hat Insights will enable analytics and additional
management capabilities on your system.
View your connected systems at https://console.redhat.com/insights

You can learn more about how to register your system
using rhc at https://red.ht/registration
Last login: Wed Jan  7 19:44:39 2026 from 10.94.0.208
[omega.uta.edu]:/home/s/sn/sn: 
```

- Mac users: **Terminal** app -> **ssh abc1234@omega.uta.edu**

Your password won't echo.

To log off from `omega`, `control-D` or type `exit` or `logoff`.

Getting around Unix

Once you login to **omega**, try the following commands:

- Issue the following commands ($ is the system prompt, do not type it) exactly as shown:

  ```
  $ cp  ~sn/.bash_profile  .
  $ source .bash_profile
  ```

  This enables control-Z in `nano` (by default it is disabled). Required once and for all.
- Play with the following commands and figure out what they do.
  - ls (directory listing)
  - ls -l (directory listing in **l**ong format)
  - ls -lt | more (one screen at one time, long format, chronological order)
  - dir (same as `ls`)
  - whoami
  - who
  - cal (calendar)
  - cal 1980
  - up/down arrow (control-P, control-N) (recalling previous commands)
  - (bring up who) and control-A, control-E, control-F, control-B, control-D
  - ls . (current directory)
  - cd .. (directory one level up)

  - pwd (present working directory)
  - cd / (goes to the top directory)
  - cd (return to your home directory)
  - mkdir MyNewFolder (create a new directory)
  - rmdir MyN(*tab*) (remove a directory)
  - nano junk.txt (create a new file)
  - cp junk.txt junk2.txt (copy)
  - mv junk.txt junknew.txt (rename)
  - rm junk.txt (delete)
  - rm *.c (do not do this)

nano/your first C program

-
  ```
  $ nano MyFirstProgram.c
  ```
- enter the following:

  ```
  #include <stdio.h>
  int main()
  {
  printf("Hello world !\n");
  return 0;
  }
  ```
- Up/down arrow, control-A, control-E, control-F, control-B, control-D, control-K.
- Try control-K and control-U (uncut)
- To save the document, control-o
- To exit from `nano` temporarily, control-Z.
- `fg` to get back to `nano`.
- control-Z again, try

  ```
  $ ls
  $ ls -lga MyFirstProgram.c
  $ cat MyFirstProgram.c
  ```
- At the prompt, type

  ```
  $ gcc MyFirstProgram.c
  $ a.out
  ```
- if `a.out` does not work, try `./a.out`.
- `fg` to get back to nano, edit and run again.
- to exit from nano, **control-x**.
- to exit from omega.uta.edu, **control-D** or type `logoff` or `exit`.

Assignment (due 1/19/2026)

Compose, execute and run the following programs and figure out what they do. Then, write C programs following the instructions: A hint for each problem is provided. The same rule as homework submission applies.

1. **Prints three blank lines followed by "Hello World."**

   Sample program:

```
#include <stdio.h>
int main()
{
/* this is a comment */
printf("Hello UTA !");
printf("\nHello World !\n");
return 0;
}
```

\ **n** sends a new line.

2. **Reads two *real* numbers from the keyboard and prints their product.**

   Sample program:

```
#include <stdio.h>
int main()
{
int a, b;  /* to declare that a and b are integer variables */
printf("Enter two integer numbers separated by space =");
scanf("%d %d", &a, &b); /* this is the way to read two integer numbers and assigned them to a and b */

printf("The sum of the two numbers is %d.\n", a+b); /* %d is for integer format*/
return 0;
}
```

   - Use `int` for integers, **float** for floating (real) numbers.
   - To enter the value of a variable from the keyboard (standard input), use **scanf()**.
   - The **scanf** function reads input from the keyboard in the format specified by (%d %d) and assigns each value to the variable followed.
   - Use **%d** for integer and **%f** for float.
   - Add an ampersand (&) before each variable when using the **scanf** function for a reason too early to discuss.
   - The **printf** function prints a string to the screen (standard output). When it encounters a format (such as **%d**), it replaces that part by the variable followed.

3. **Reads a *real* number and outputs its inverse. If 0 is read, writes a warning message** and quit.

   Sample program:

```
#include <stdio.h>
int main()
{
float a;
printf("Enter a number = ");scanf("%f", &a);

if (a==0.0) printf("You entered 0.\n");
 else printf("You entered a number other than 0.\n");
return 0;
}
```

4. **Reads a *real* number, *x*, and outputs its sine, i.e. sin *x*. You need to use `math.h` and the `-lm` compile option.**

   Sample program:

```
#include <stdio.h>
#include <math.h>
int main()
{
float x;
printf("Enter a number ="); scanf("%f", &x);
printf("x= %f exp(x)=%f\n",x, exp(x));
return 0;
}
```

   Sample run (note the -lm option):

```
$ gcc -lm myprogram.c
$ a.out
```

File translated from TEX by TTH, version 4.03.
On 09 Jan 2026, 14:22.