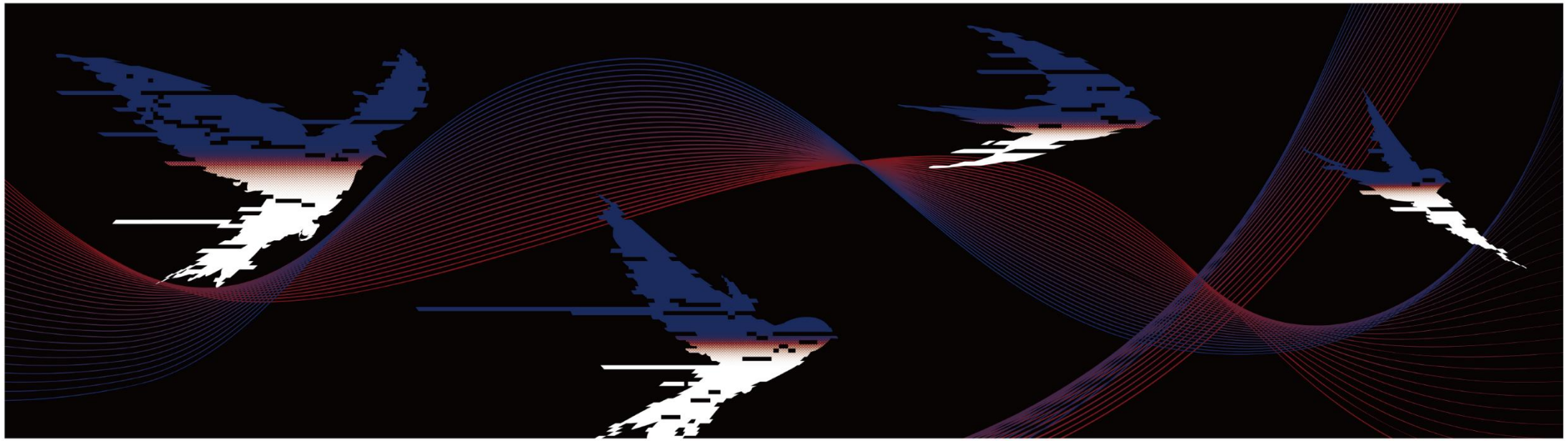
Introduction to TSUBAME (Linux basics)



Contents

- TSUBAME4 Overview
- Introduction to Linux
- File operation
- Common Linux Commands
- Running Jobs on Compute Nodes
- User environment

TSUBAME4 Overview

- System Architecture
- Commercial applications
- Available Services
- Getting Started
- T4 Web page

TSUBAME4

- TSUBAME4 is a supercomputer that started operation on April 1, 2024.

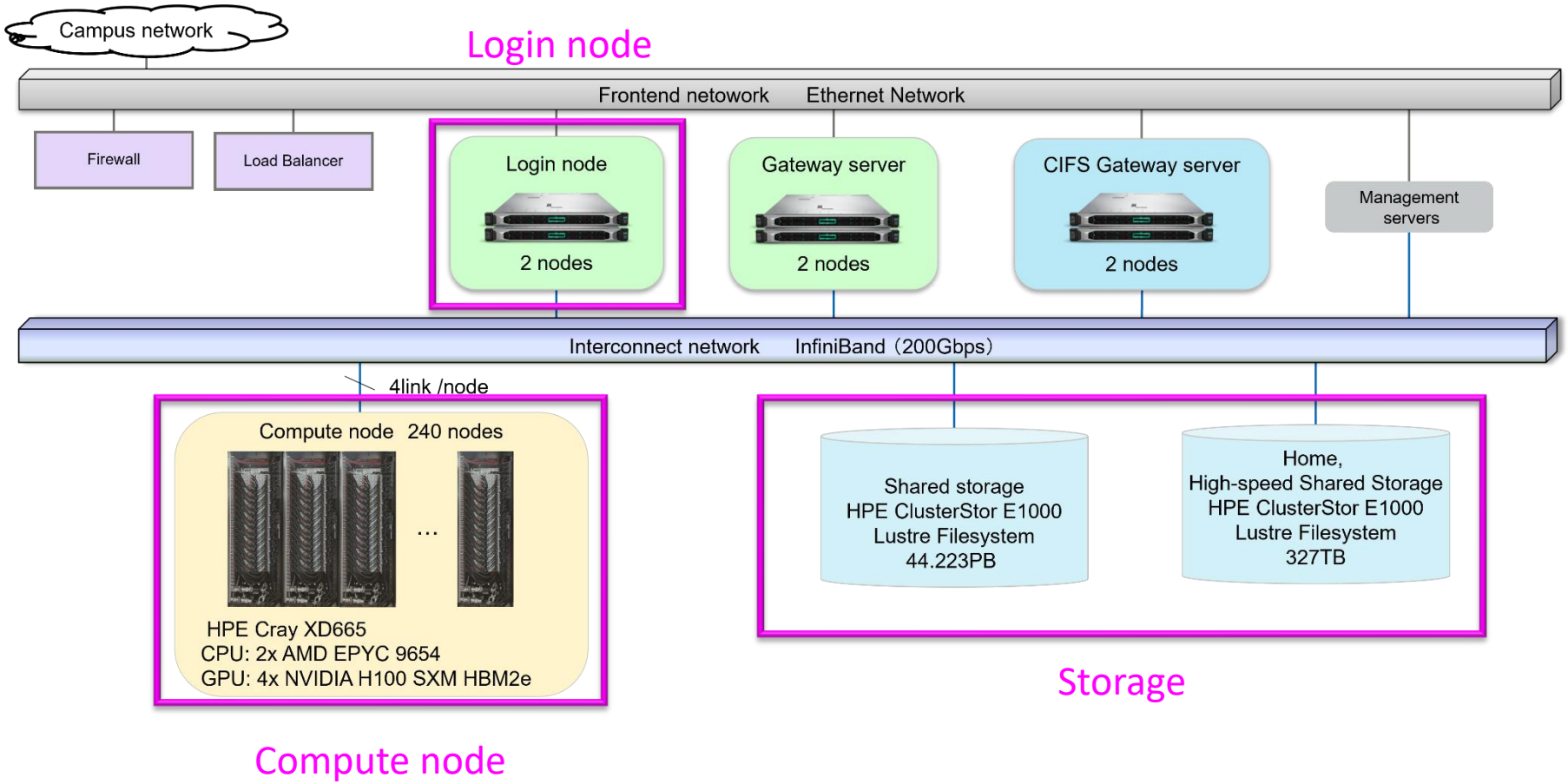
- 960 NVIDIA H100 GPUs
- Theoretical peak performance
  - FP64 (double precision): 66.8 PFLOPS
  - FP16 (half precision): 952 PFLOPS
- Total memory capacity：180 TiB
- Total SSD capacity: 327 TB
- Total HDD capacity: 44.2 PB
- 200 Gbps High-speed Infiniband network

- TSUBAME4 also provides web-based access to computing resources via Open OnDemand.
  - Users can run jobs ranging from small CPU jobs to large-scale multi-GPU jobs.
  - Standard SSH access is also available.

- Programs developed on TSUBAME3 can be used on TSUBAME4 without modification.
  - Architecture: x86_64 CPUs + NVIDIA GPUs + Linux OS

*1 TiB = 2^{40} B = 1,099,511,627,776 B

T4 System architecture



System details - Compute node

- HPE Cray XD665 — 240 nodes

Item	Specification
CPU	2 x AMD EPYC 9654 CPUs
# of cores/threads	192 CPU cores / 384 threads per node
Memory size	768 GiB memory
GPU	4 x NVIDIA H100 GPUs
SSD	1.92TB NVMe SSD
Interconnect	4 × 200 Gbps InfiniBand interfaces

System details - Storage and SW

- Storage

Areas	Mount point	Capacity	Filesystem
Home directories (High-speed SSD-based storage)	/gs/fs /home	372TB	Lustre
Shared applications (Large-capacity HDD-based storage)	/gs/bs /apps	44.2PB	Lustre
Local NVMe SSD scratch space on each compute node	/local	1.92TB	xfs

- Software
  - Red Hat Enterprise Linux Server 9.4
  - Commercial applications available (listed on the next page)

Commercial applications

Software	description
ANSYS	Analysis Software
ANSYS/Fluent	Analysis Software
ANSYS/LS-DYNA	Analysis Software
ABAQUS/ABAQUS CAE	Analysis Software
Gaussian	Quantum chemistry calculation program
GaussView	Quantum chemistry calculation program Pre-Post tool
AMBER	Molecular dynamics calculation program
Materials Studio	Chemical Simulation Software
Discovery Studio	Chemical Simulation Software
Mathematica	Mathematical Processing Software
COMSOL	Analysis Software
Schrodinger	Chemical Simulation Software
MATLAB	Numerical calculation software
Arm Forge	Debugger
Intel oneAPI Compiler	Compiler, Development tool
NVIDIA HPC SDK Compiler	Compiler, Development tool

Applications must be enabled before use.

Activation can be managed through the T4 portal using TSUBAME points.

Application environments are configured using the module command

Some applications are available only to on-campus users.

Computing & Storage Services

- Compute Nodes
  - 240 HPE Cray XD665 compute nodes are available. See a later section for details on compute node usage.

- Storage
  - Home Directory (25 GiB, free)
    - Personal directory: /home/[0-9]/<username>
    - Accessible from all nodes in the system
    - Location can be checked with: echo ${HOME}
  - Work Directory (100 GiB, free)
    - Personal directory: /work/[0-9]/<username>
    - Location can be checked with: echo ${HOME/home/work}
  - High-Speed / Large-Capacity Storage (paid)
    - Shared group storage based on the Lustre file system
    - Available while the subscription is active
  - Scratch Space (available only during job execution)
    - Local scratch storage based on SSDs
    - Automatically removed after the job finishes

Paid Services

- Run jobs on compute node (pay-as-you-go)
- Compute node reservation
- 1-month node reservation (flat-rate system)
- Commercial applications used in T4 system (monthly payment)
- Parts of Commercial applications used in campus (annual payment)
- Group Disk

- TSUBAME points are required to use paid services.
- TSUBAME points are allocated based on your research project budget.
- TSUBAME points expire at the end of the fiscal year.

Getting Started

[note] Participants in this course need to configure 1 and 2.
TSUBAME point is necessary to submit jobs, use group disks and use commercial application.

1. Get a TSUBAME account (See https://www.t4.cii.isct.ac.jp/en/getting-account)
2. Create SSH key pair in local PC and upload the public key
3. Create group [by group administrator]
4. Configure group
   - Register budget code [by group admin.]
   - Purchase points [by group admin. or sub admin.]
   - Add users to the group [by group admin. and users]
   - Grant permission to users [by group admin. or sub admin.]
   - Configure group disks [by group admin. or auth. users]
   - Apply for using Apps. (activation) [by group admin. or auth. users]
   - <Another configuration if you need>
5. Submit jobs

Setup at TSUBAME4 potal

Usage of compute node

- There are two main ways to use compute nodes.

1. SSH Access: (Command-line interface (CLI))
2. Open OnDemand: Web Interface

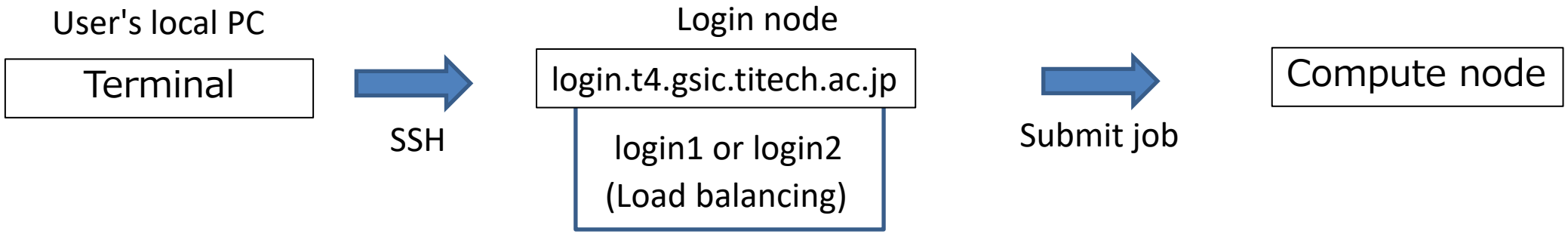
- In this course, CLI use with SSH will be introduced.

Usage of compute node (SSH)

- Login to login node (SSH key authentication)
  - Use the ssh command to connect to login.t4.gsic.titech.ac.jp.
  - You will automatically be logged in to one of the login nodes via load balancing.

※ Do not run computationally intensive jobs on login nodes. If the monitoring system detects such jobs, they will be terminated automatically. Login nodes are not intended for computation.

※Upload SSH public key to server (via T4 portal) in advance. After upload SSH key, users finally can access login node. See TSUBAME Portal User's Guide.



TSUBAME information

- TSUBAME4 Computing Services (Homepage)
  - https://www.t4.cii.isct.ac.jp/en

- TSUBAME4 portal
  - https://portal.t4.gsic.titech.ac.jp/ptl/

    For user/group setting (e.g. SSH pubkey registration, point purchase, node reservation)
- X (formerly known as Twitter)
  - @TSUBAME_sc

    The latest news will be announced on X and T4 Homepage.

- If you have questions, please contact the TSUBAME support team through the "Contact Us" page..
  - https://www.t4.cii.isct.ac.jp/en/contact-t4

Introduction to Linux

- Overview of Linux
- Terminal
- Use terminal on Windows/Mac
- Remote login
- SSH key authentication
- Login to TSUBAME4
- Logout

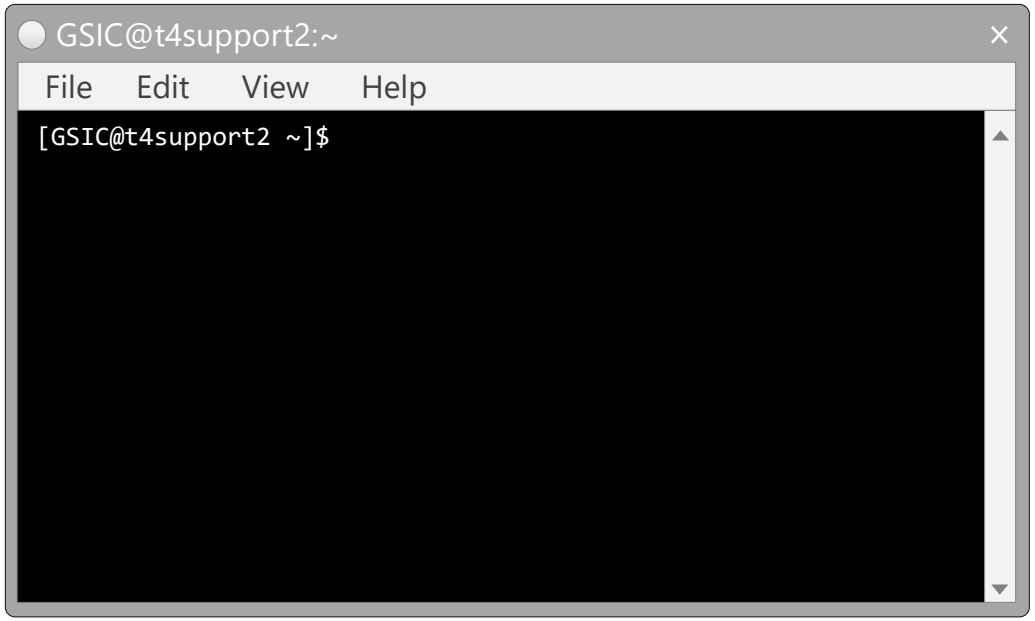
Overview of Linux

- Linux is a family of open-source Unix-like operating systems based on the Linux kernel.
- Linux is a multitasking/multiuser Operating System.
- Linux distributions
  - Debian (Ubuntu)
  - Slackware Linux
  - SUSE Linux Enterprise Server (SLES)
  - Red Hat Enterprise Linux (RHEL)

  …
- Authentication
  - Local login
    - username / password
  - Remote login
    - username / password
    - key-based authentication <-- TSUBAME4 supports this authentication.

Terminal

- A text-based interface for interacting with a Linux system.
- Users enter commands through a terminal to manage files, run programs, and access remote systems.
- The line which has a symbol such as “%”, “$” and “>” is called prompt. (In the example below, $ is displayed.)
- Type a command on this line.



Windows Terminal Software

- There are various terminal emulators for Windows.
  - Cygwin, PuTTY, Tera Term, Rlogin, MobaXterm, WSL ....

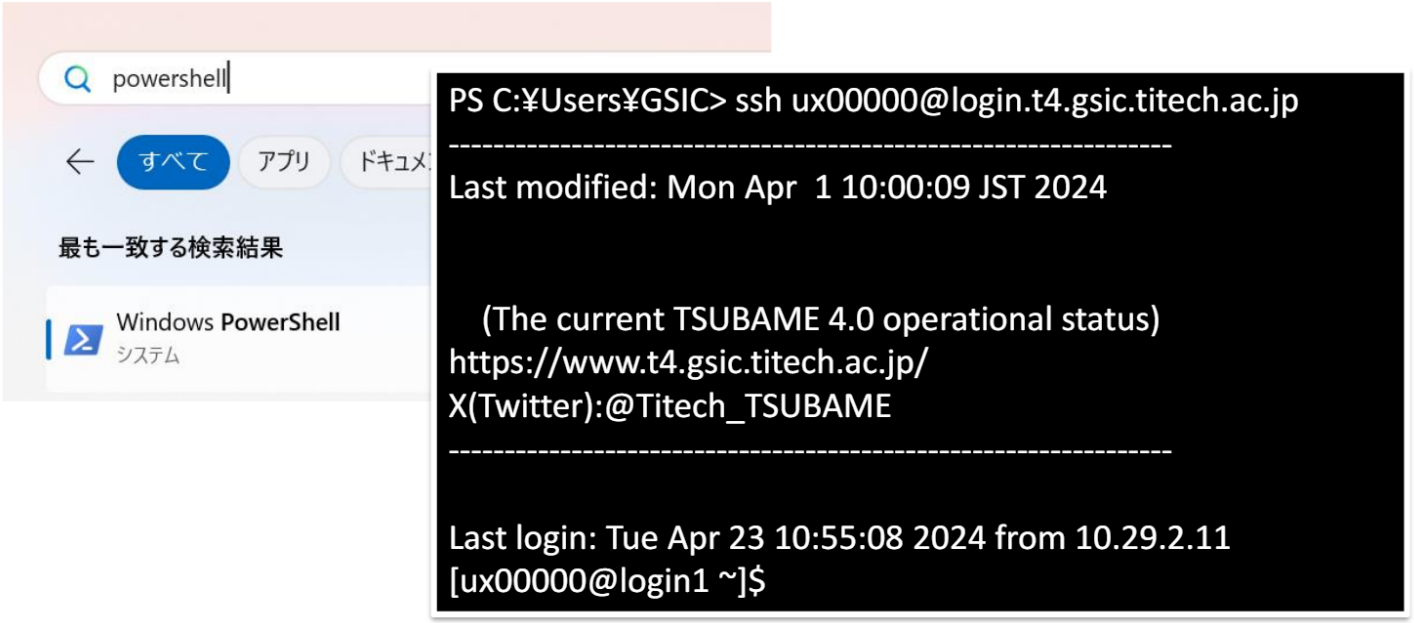
- MobaXterm is a popular choice because it provides SSH, file transfer, and X11 forwarding in a single application.

- Choose the terminal application that best suits your needs.

See also Available SSH client on Windows

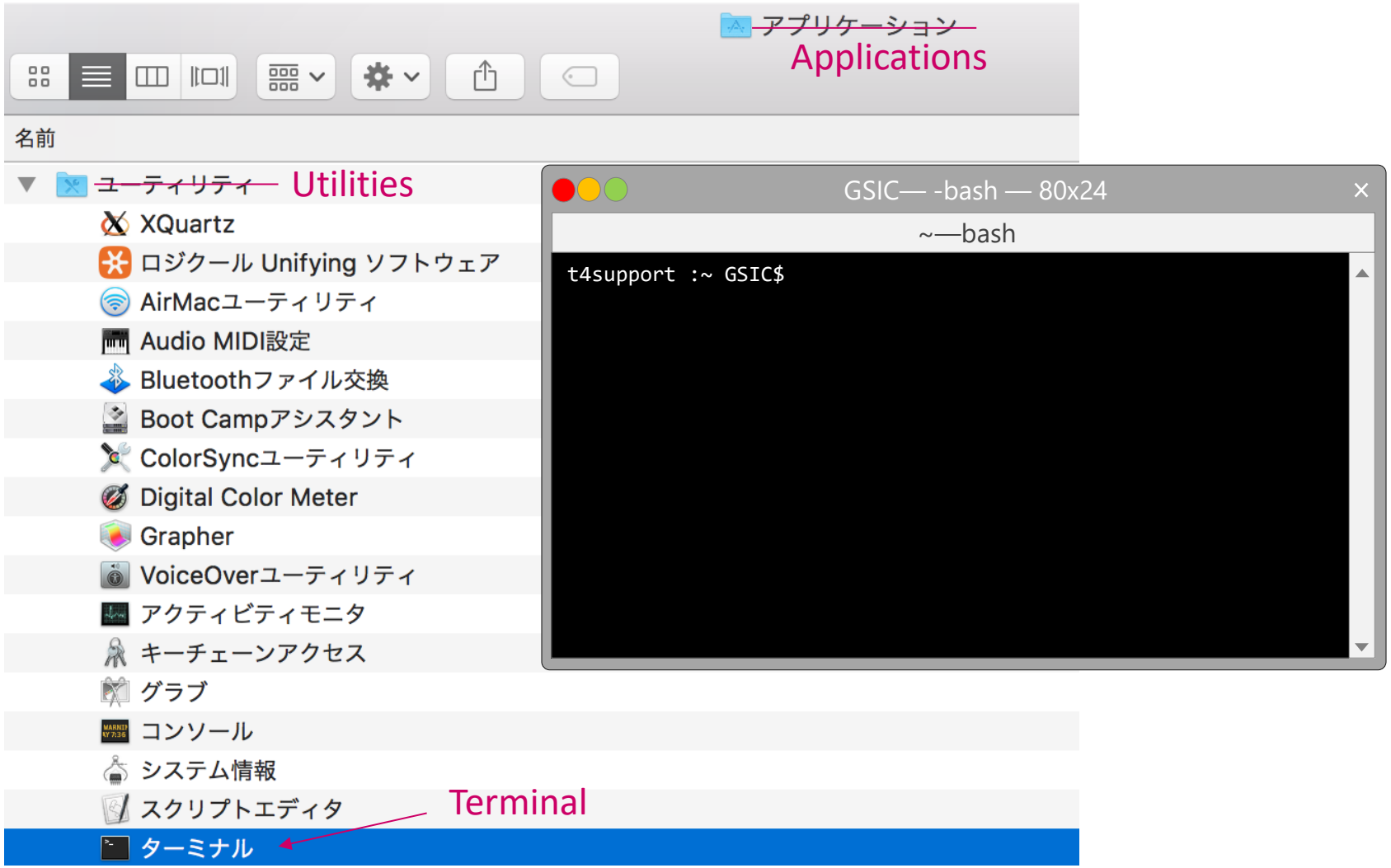
PowerShell

- Open the Start Menu
- Search for "PowerShell"
- Launch Windows PowerShell



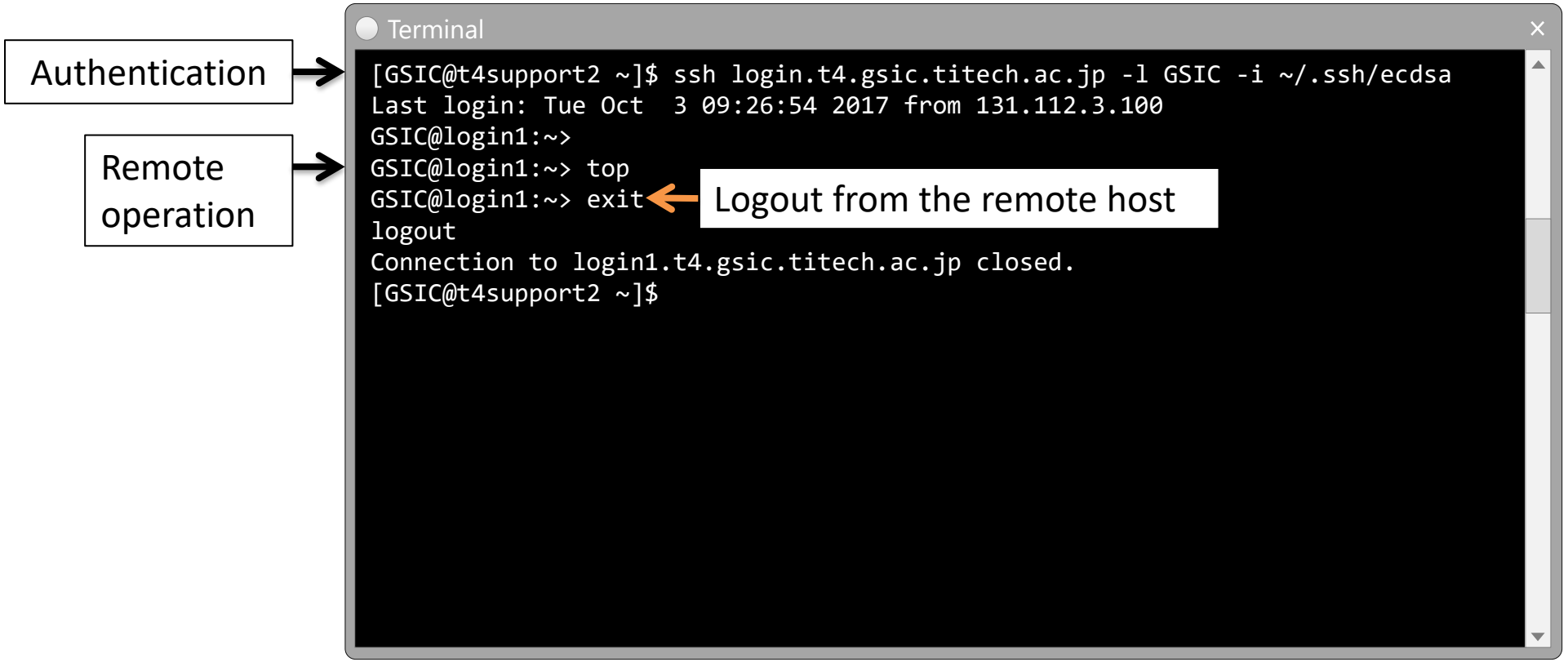
Mac OS Terminal

- Applications >Utilities >Terminal.app



Remote login

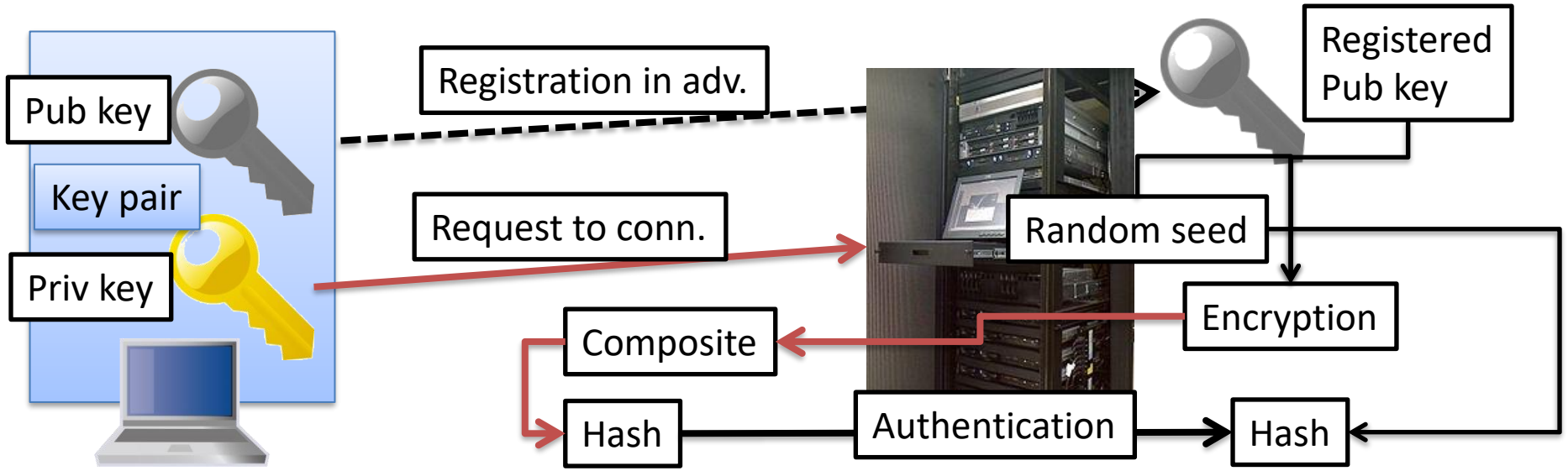
- Remote host operation on a local host
- Commands are *telnet*, *rlogin*, *ssh* and so on.
- In TSUBAME, only SSH public key authentication is supported for security reasons.



SSH key authentication

- SSH public-key authentication uses a pair of keys:
  - Public key → registered on the server
  - Private key → on your local computer

- Anyone who obtains your private key may be able to access your account.
- We strongly recommend protecting your private key with a passphrase.



Creating SSH key pair

- Use ssh-keygen command

See https://www.t4.cii.isct.ac.jp/docs/faq.en/general/#keypair

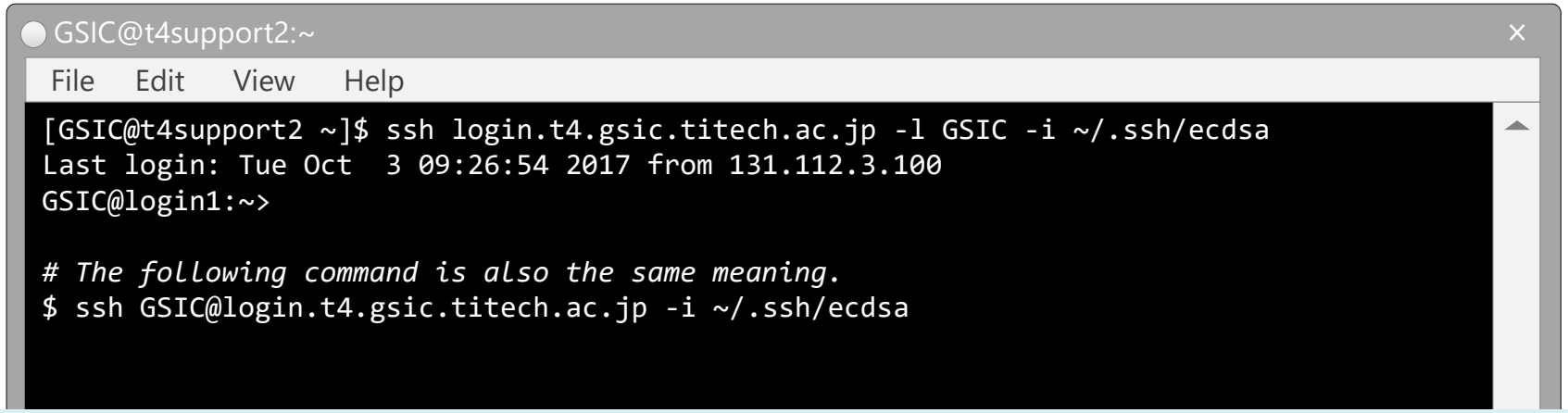
Upload public key

Upload public key to TSUBAME via T4 poral.

- https://portal.t4.gsic.titech.ac.jp/ptl/user/sshPublicKey?lang=en
- Check the text of public key (.pub), copy and paste it to the T4 potal page.

Exercise (Login to TSUBAME4)

- Input ssh command on a prompt to log in to TSUBAME4 with specifying a ssh key as shown below.
- Successfully logged in, a prompt *username@loginX* appears.



You don't have to use your private key if you want to use TSUBAME from a Computer room's PC in campus. (password authentication)

Therefore, please execute ssh command without -i option while this seminar.

```
$ ssh login.t4.gsic.titech.ac.jp -l username
```

Logout

- Perform log out operation to finish your work.
- Check the followings before log out.
  - Data arrangement
  - Process (Check no program is running on the terminal)

- Log out operation
  - Type Ctrl and d keys at the same time (Ctrl-d)
  - Type logout -> Enter
  - Type exit -> Enter

Tips

- It is better to configure the terminal preference when you log in to TSUBAME to avoid the disconnect.

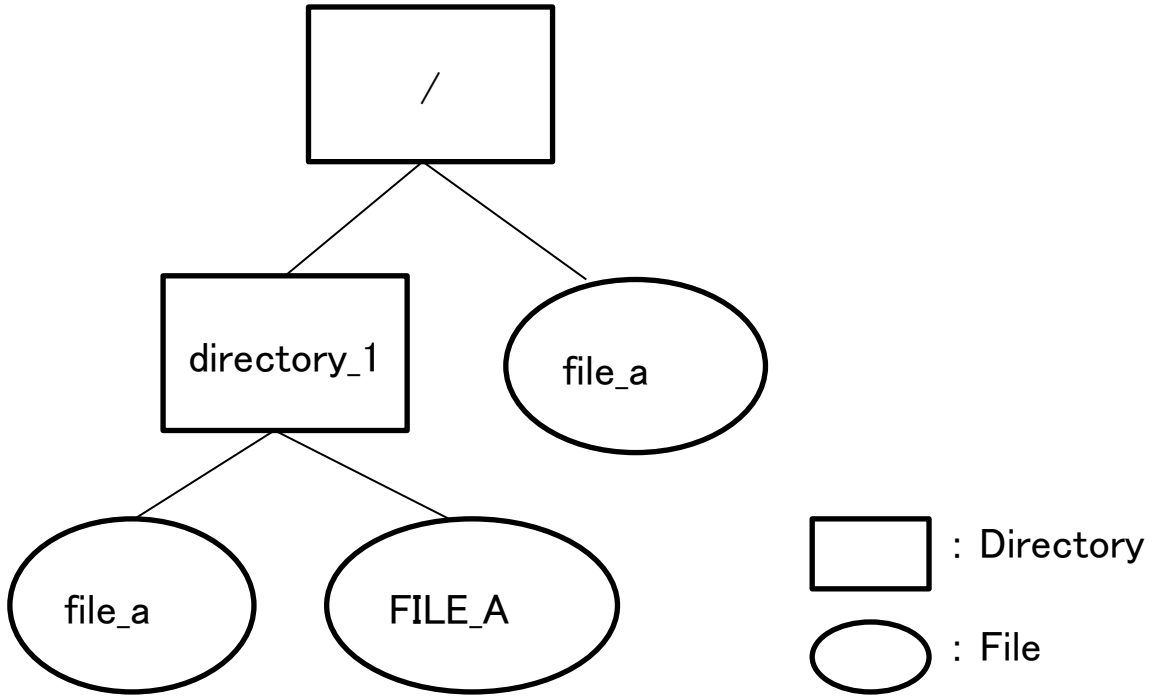
- Discribe the following parameteres in ~/.ssh/config if you use Mac, Cygwin, Linux and so on.

File Operation

- Filesystem
- File specification
- Information display commands
- Symbols
- Special characters
- File permission and attribute
- Newline difference amang OS
- Compression and extraction

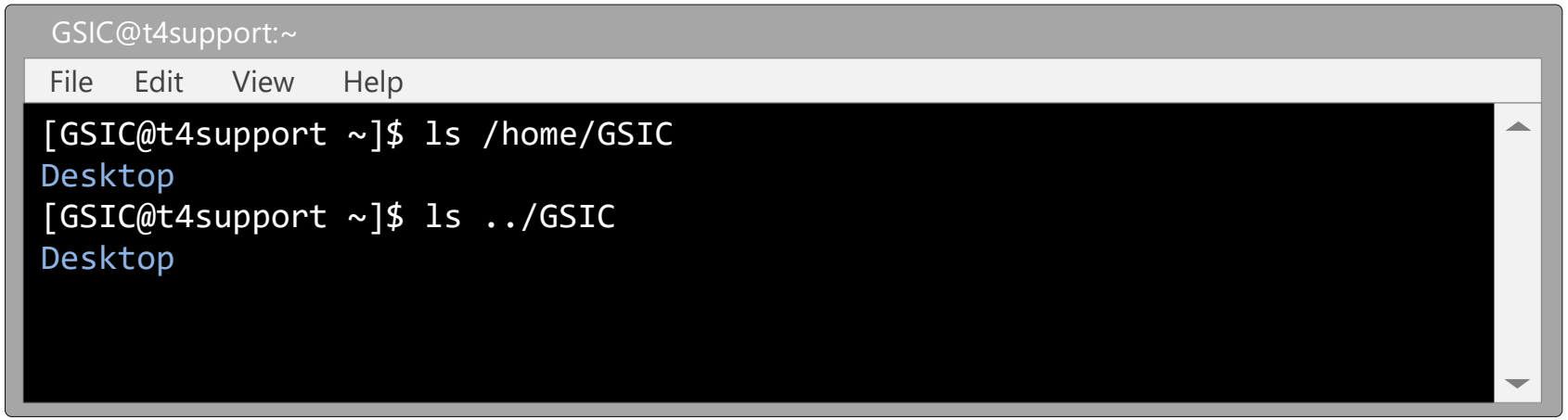
Filesystem

- The Linux file system has a hierarchical structure.
- Files are organized within directories.
- The top of the hierarchy is called the root directory (/).



File Paths

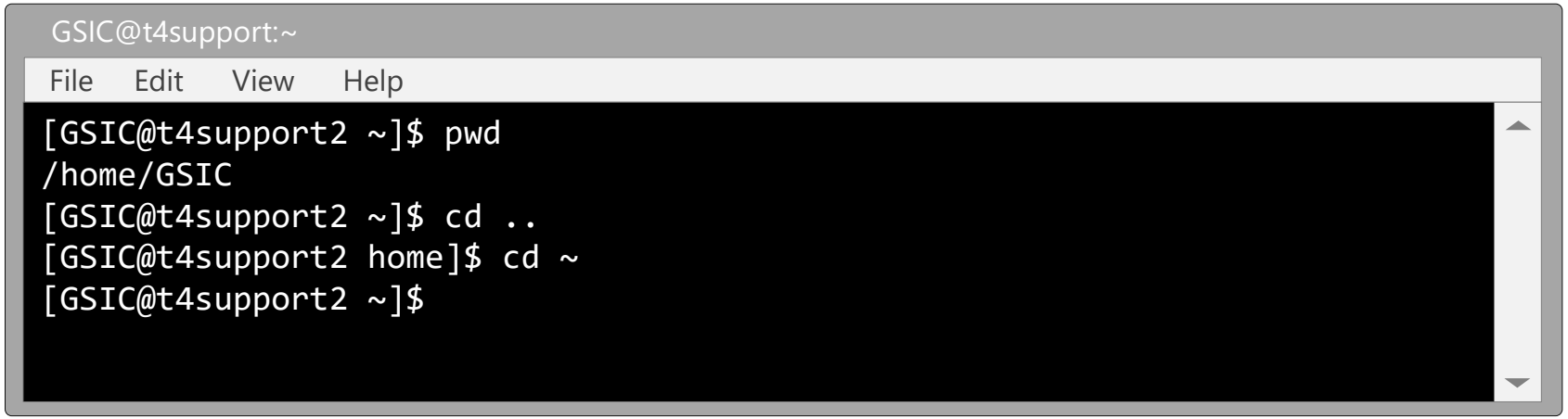
- The top of a hierarchical structure of a file system is called “root”, represented as “/”.
- How to access files
  - Absolute path: starts from (/)
  - Relative path: starts from the current directory



Special Directory Symbols

- Command to confirm directory: pwd

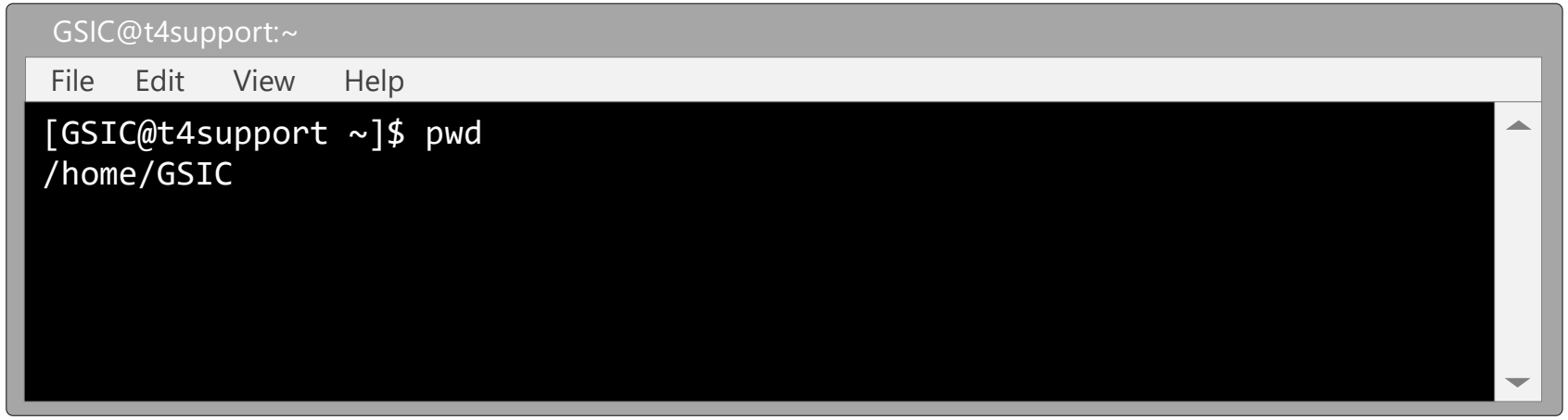
- . Current directory
- .. Parent directory
- ~ Home directory



***Use . (dot) to represent the current directory (current position)**

Information display commands

- pwd (Display the current working directory)



- ls (List files in current or specified directory)

Special character patterns

- Special character patterns can be used to specify file names.

*Metacharacter	Function	Usage
*	all character strings	ls *
?	one of any character	ls ?
[character string]	one of character strings	ls [bc]
[character1-character2]	one of the characters between character 1 and character 2	ls b[a-c]d

*Metacharacter

Special characters are also known as metacharacters. Metacharacters are symbols which do not have meaning on their own, however these become meaningful as a whole when combined with other characters.

File permission and attribute

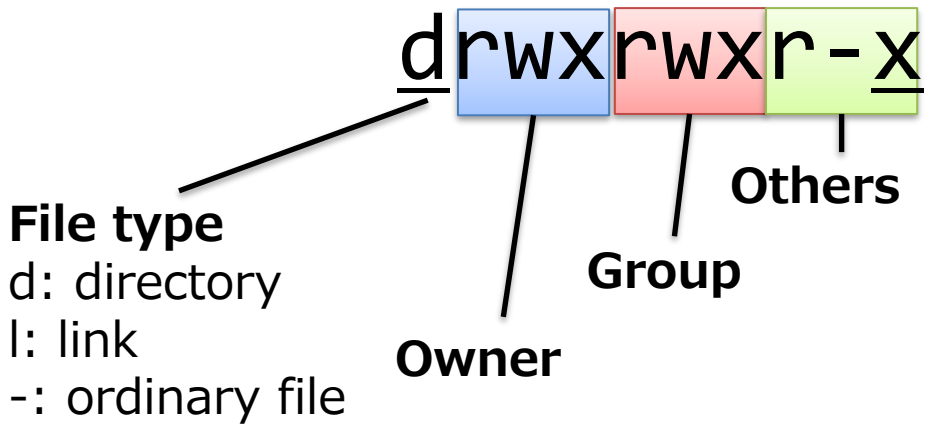
- ls -l

GSIC@g3support2:~

File Edit View Help

```
[GSIC@t4support2 hogehoge]$ ls -l
合計 0
drwxrwxr-x. 2 GSIC users 6 10月  3 15:35 hoge
-rw-rw-r--. 1 GSIC users 0 10月  3 15:35 hogedoc
```

Attribute | Owner, Group | Timestamp | File name



Permission
r: read
w: write
x: execute
-: unauthorized

Commands for file management

- Create directory % mkdir <file>
- Remove directory % rmdir <file>
- Change file attribute % chmod 755 <file>

*755 is called as bit representation, and it represents access rights for owner, group and others.

0	---	unauthorized
1	--x	execute
2	-w-	write
3	-wx	
4	r--	read
5	r-x	
6	rw-	
7	rwx	full access

(Operation of administrator is as follows)

- Change owner # chown necapps <file>
- Change group # chgrp procon <file>

Character Encoding and Line Endings

- Different operating systems use different line endings:
  - Linux ¥n LF(line feed)
  - Macintosh ¥r CR(carriage return)
  - Windows/Dos ¥r¥n CRLF

- [note] A text file contains Win/Dos-type newline cannot be read on Unix/linux system.

- Display code for 2-bite characters (e.g. Japanese)
  - Unix/Linux UTF-8/EUC
  - Macintosh/Windows UTF-8/S-JIS

- Conversion through nkf command
  - nkf -Lu abc_*crlf.sh* > abc_*fl.sh*

Compression and extraction

- Compression

  gzip atom45.tar → atom45.tar.gz
  zip atom45.zip atom45 → atom45.zip
  lha a text.lzh *.txt → text.lzh
  tar czvf atom45.tgz atom45 → atom45.tgz
  tar cjf smpl.tar.bz2 smpldir → smpl.tar.bz2
  bzip2 sample.txt → sample.txt.bz2

- Extraction

  zcat atom45.tar.Z | tar -xvf -
  tar jxf sample.tar.bz2
  gzip -d atom45.tar.gz → atom45.tar
  unzip book2nd.zip
  lha e text.lzh
  tar xzvf atom45.tgz
  bzip2 -d sample.txt.bz2 → sample.txt

Various commands

- Frequently used commands
- Commands for file operation
- Alias function
- Text editor (vi)
- Usage of commands
- Online manual
- Command concatenation

Frequently used commands

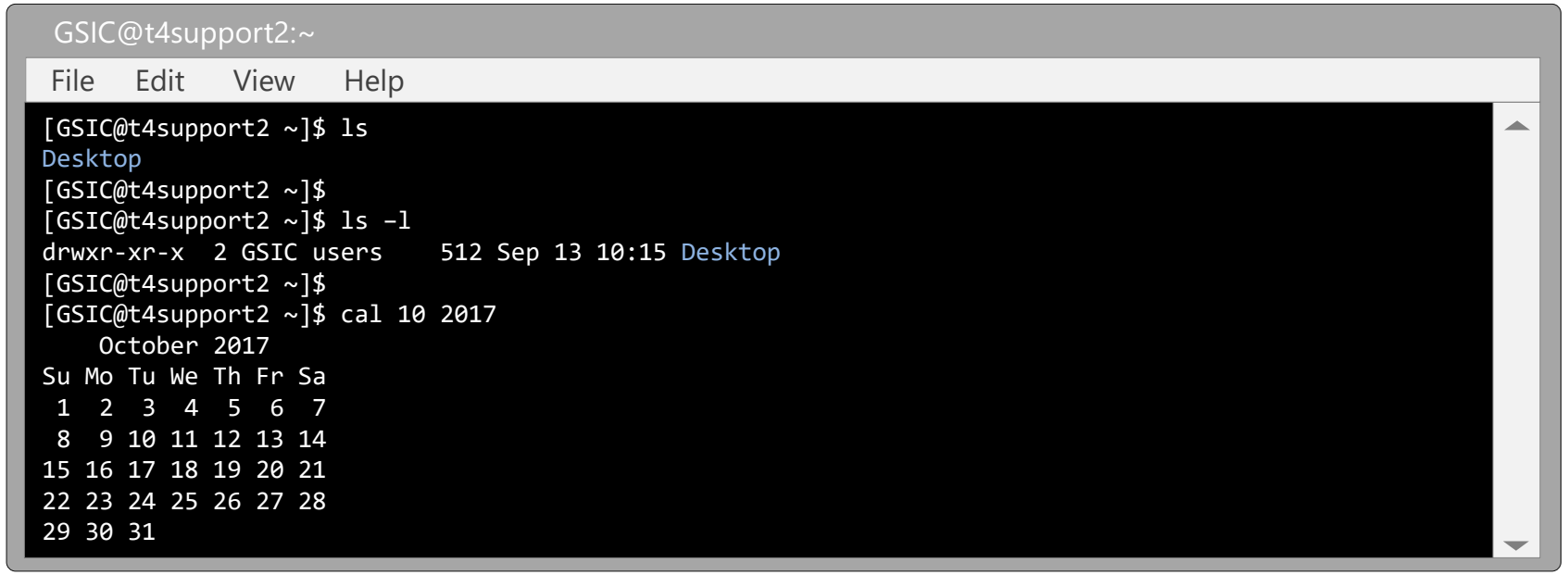
- ssh
- exit
- mkdir
- rmdir
- chmod
- chown
- chgrp
- nkf
- cd
- cp
- mv
- rm
- pwd

- ls
- vi
- emacs
- view
- tail
- cat ,more ,less
- find
- file
- grep
- diff ,sdiff
- man

Command Usage

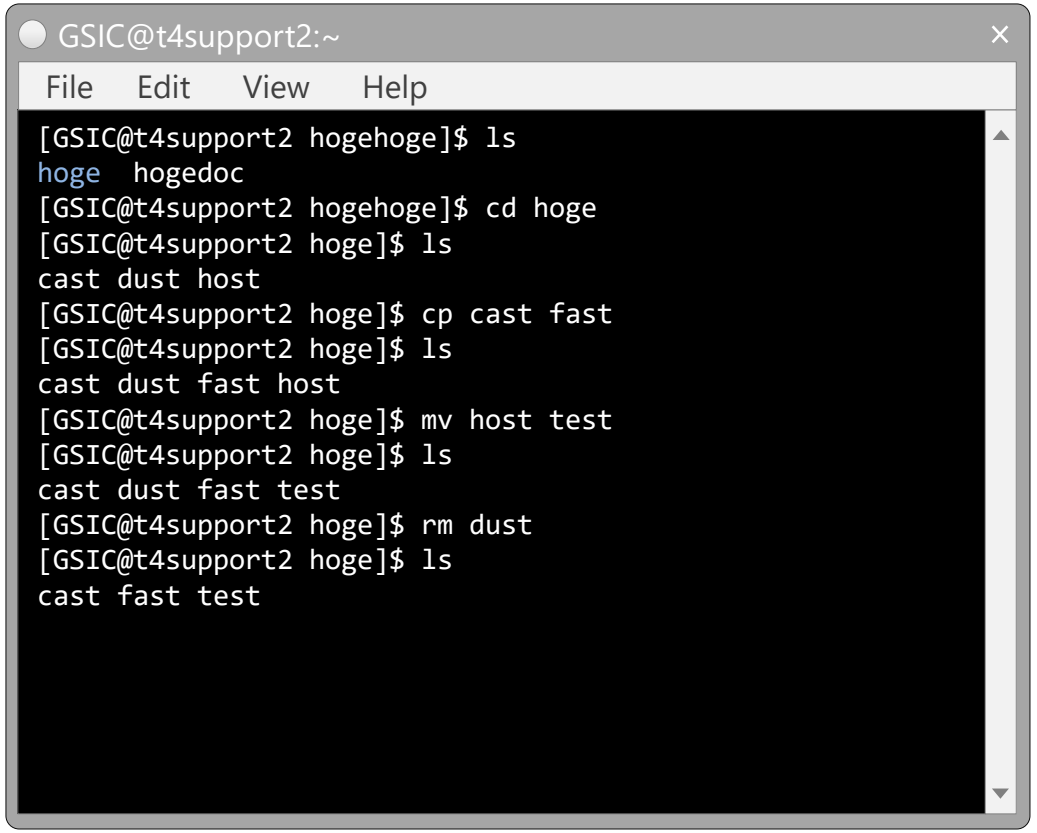
Usage

- Execute a command without any option
- Execute a command with options or arguments
- It is possible to combine multiple commands.



Commands for file operation

- cd
  change directory
- cp
  copy
  copy a file or directory
- mv
  move
  move a file or directory
- rm
  remove
  remove a file or directory

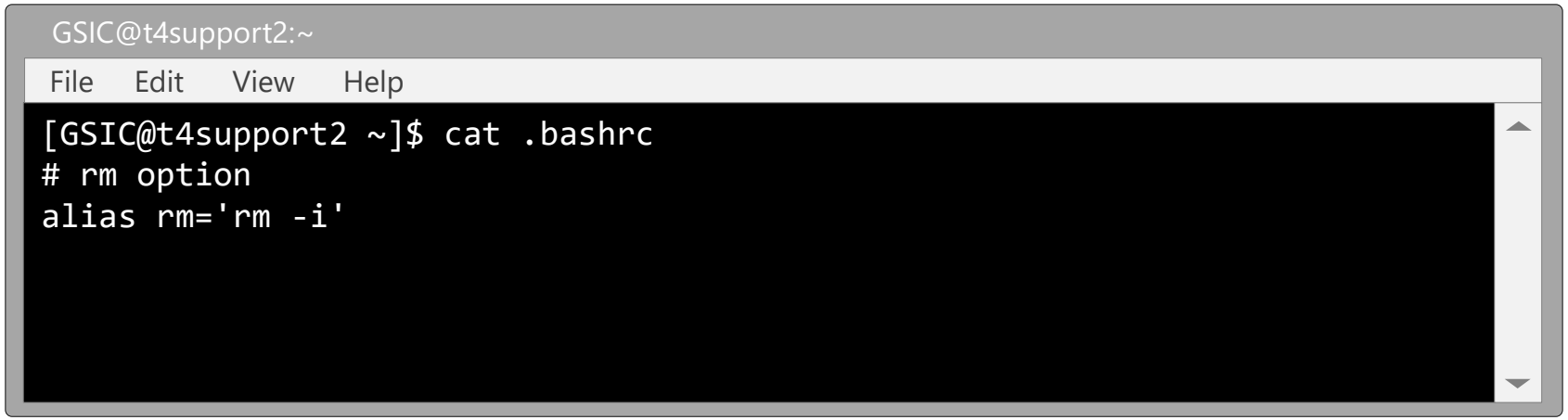


*In cp, mv, rm commands, users are prompted for confirmation by –i option.

*There is no command to restore files once they have been deleted.

Alias function

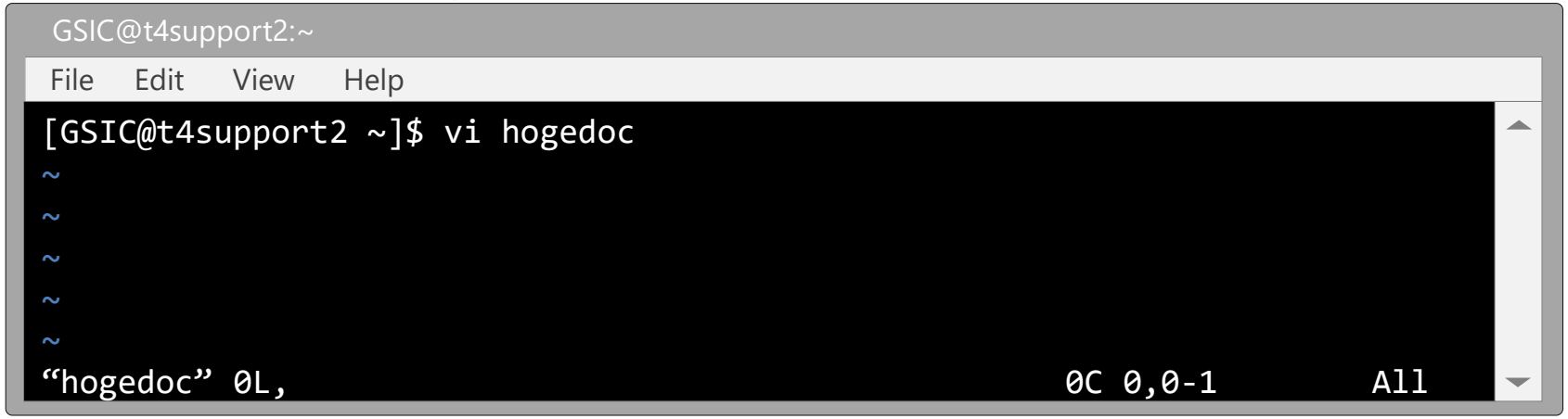
- Create shortcuts for frequently used commands.



By writing this way in .bashrc, loss of file by mistake can be prevented.
By mistake, if space is given such as rm * .txt , then all files in current directry will be deleted.
By setting the alias, users are prompted for confirmation before removing by -i option.

Text editor (vi)

It is a default editor in Linux systems.

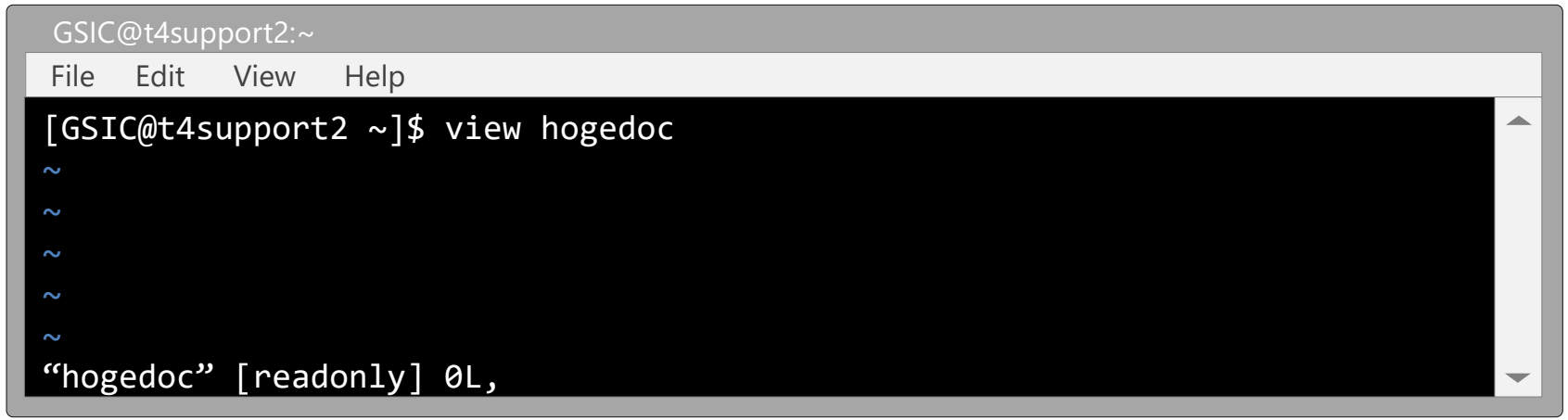


i	Enter insert mode
ESC	Back to Standby mode (Command mode)
x	Delete one character
dd	delete one line
:wq	Save and exit editor
:q!	Wipe out all edits and quit (exit vi editor)
u	Undo the last change to the file

On standby mode, move using h, j, k, l, which is the function of ←, ↓, ↑, →, respectively.

view

- Open a file in read-only mode.
- Equivalent to starting vim in read-only mode.



tail

- Displays last part of output to standard output

GSIC@t4support2:~

File Edit View Help

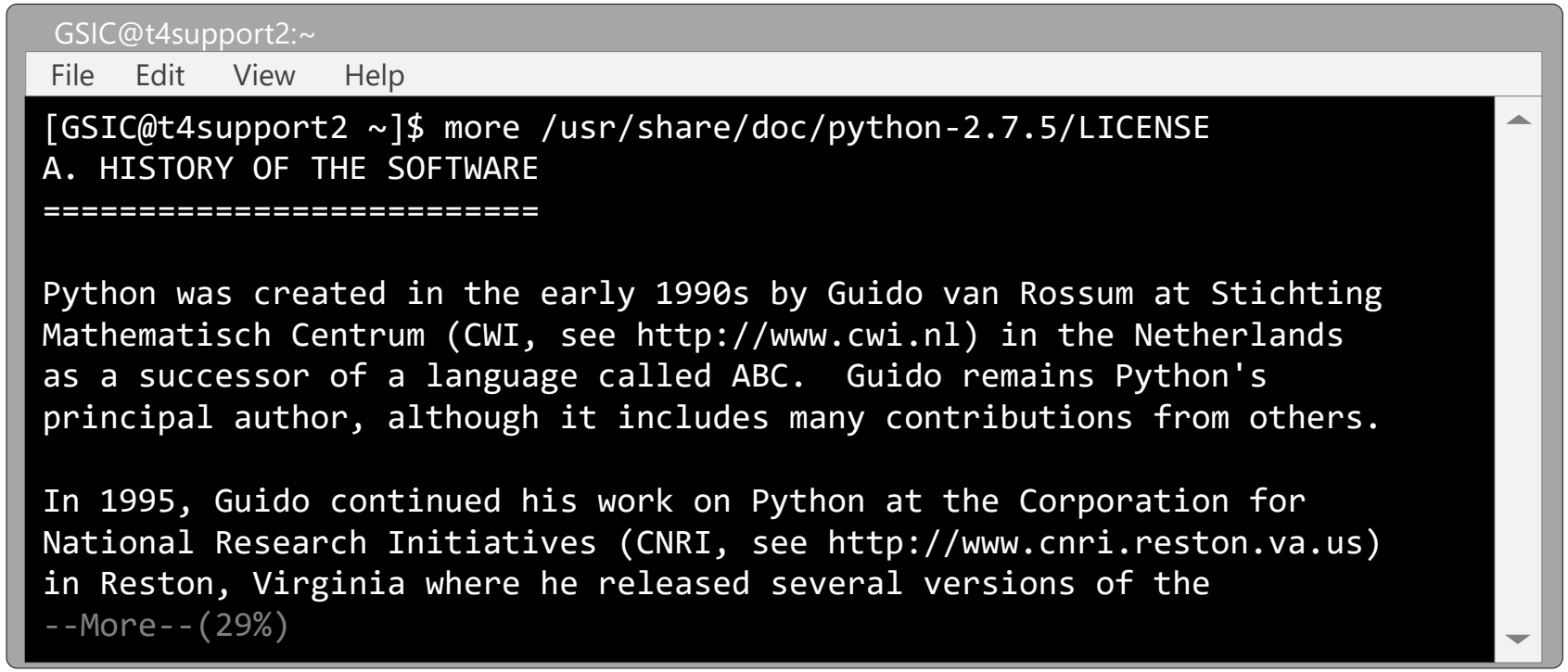
```
[GSIC@t4support2 ~]$ tail -5 /usr/share/doc/python-2.7.5/LICENSE
FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE
FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
```

It is useful to output the calculation result to a file and periodically check the progress with the tail command.

Please avoid running text output and tail commands successively through a program.
(in order to avoid the heavy load on the machine)

cat, more and less

- Commands for viewing file contents:
  - cat : display the entire contents of a file
  - more : view a file page by page
  - less : view a file interactively, with scrolling support
- Press the Space key to move to the next page.



find, which and whereis

Search for files and directories.

- Related commands:
  - which — Locate an executable command
  - whereis — Locate binaries, source files, and manual pages

GSIC@t4support2:~

File Edit View Help

```
[GSIC@t4support2 ~]$ find . -name hoge –print
./hoge

[GSIC@t4support2 ~]$ which python
/usr/bin/python

[GSIC@t4support2 ~]$ whereis python
python: /usr/bin/python /usr/bin/python3.4 /usr/bin/python3.4m
/usr/bin/python2.7 /usr/bin/python2.7-config /usr/bin/python3.4-config
/usr/bin/python3.4m-config /usr/lib/python3.4 /usr/lib/python2.7
/usr/lib64/python3.4 /usr/lib64/python /usr/lib64/python2.7
/usr/include/python3.4m /usr/include/python2.7 /usr/include/python
/usr/share/man/man1/python.1.gz
```

file

- Display the type of a file based on its contents

GSIC@t4support2:~

File Edit View Help

```
[GSIC@t4support2 ~]$ file hoge
hoge: directory

[GSIC@t4support2 ~]$cd hoge
[GSIC@t4support2 hoge]$ file hogedoc
hogedoc: empty

[GSIC@t4support2 hoge]$ file hogedoc2
hogedoc: ASCII text
```

grep

- Search for text patterns
- Convenient to search strings in standard output (with pipe)

  It is useful if this command is executed before performing character string search such as vi editor and more command.

GSIC@t4support2:~

File Edit View Help

```
[GSIC@t4support2 examples]$ grep mpi *.c
connectivity_c.c:#include <mpi.h>
hello_c.c:#include "mpi.h“

[GSIC@t4support2 examples]$ grep mpi *.c
connectivity_c.c:#include <mpi.h>
connectivity_c.c:    MPI_Status  status;
(snip)
hello_c.c:#include "mpi.h“
hello_c.c:    char version[MPI_MAX_LIBRARY_VERSION_STRING];
(snip)
[GSIC@t4support2 hoge]$ grep -i mpi *.c | more
(snip)
```

diff

Compare two files.

- diff

GSIC@t4support2:~

File Edit View Help

```
[GSIC@t4support2 ~]$ diff hello_c.c hello_cxx.cc
18,20c18,20
<     MPI_Init(&argc, &argv);
<     MPI_Comm_rank(MPI_COMM_WORLD, &rank);
<     MPI_Comm_size(MPI_COMM_WORLD, &size);
---
>     MPI::Init();
>     rank = MPI::COMM_WORLD.Get_rank();
>     size = MPI::COMM_WORLD.Get_size();
```

- sdiff

GSIC@t4support2:~

File Edit View Help

```
[GSIC@t4support2 ~]$ sdiff hello_c.c hello_cxx.cc
    MPI_Init(&argc, &argv);                         |     MPI::Init();
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);           |     rank = MPI::COMM_WORLD.Get_rank();
    MPI_Comm_size(MPI_COMM_WORLD, &size);           |     size = MPI::COMM_WORLD.Get_size();
    MPI_Get_library_version(version, &len);               MPI_Get_library_version(version, &len);
```

man

- man command name/file name
- man -k keyword

GSIC@t4support2:~

File Edit View Help

```
[GSIC@t4support2 examples]$ man ls
Man: find all matching manual pages (set MAN_POSIXLY_CORRECT to avoid this)
 * ls (1)
   ls (1p)
Man: What manual page do you want?
Man:
NAME
       ls - list directory contents
SYNOPSIS
       ls [OPTION]... [FILE]...
DESCRIPTION
       List information about the FILEs (the current directory by default).  Sort
entries alphabetically if none of -cftuvSUX nor --sort is specified.
       Mandatory arguments to long options are mandatory for short options too.
```

POSIX
[Portable Operating System Interface for UNIX]

A set of standard operating system interfaces based on the UNIX, specified by IEEE.
(extracted from e-word)

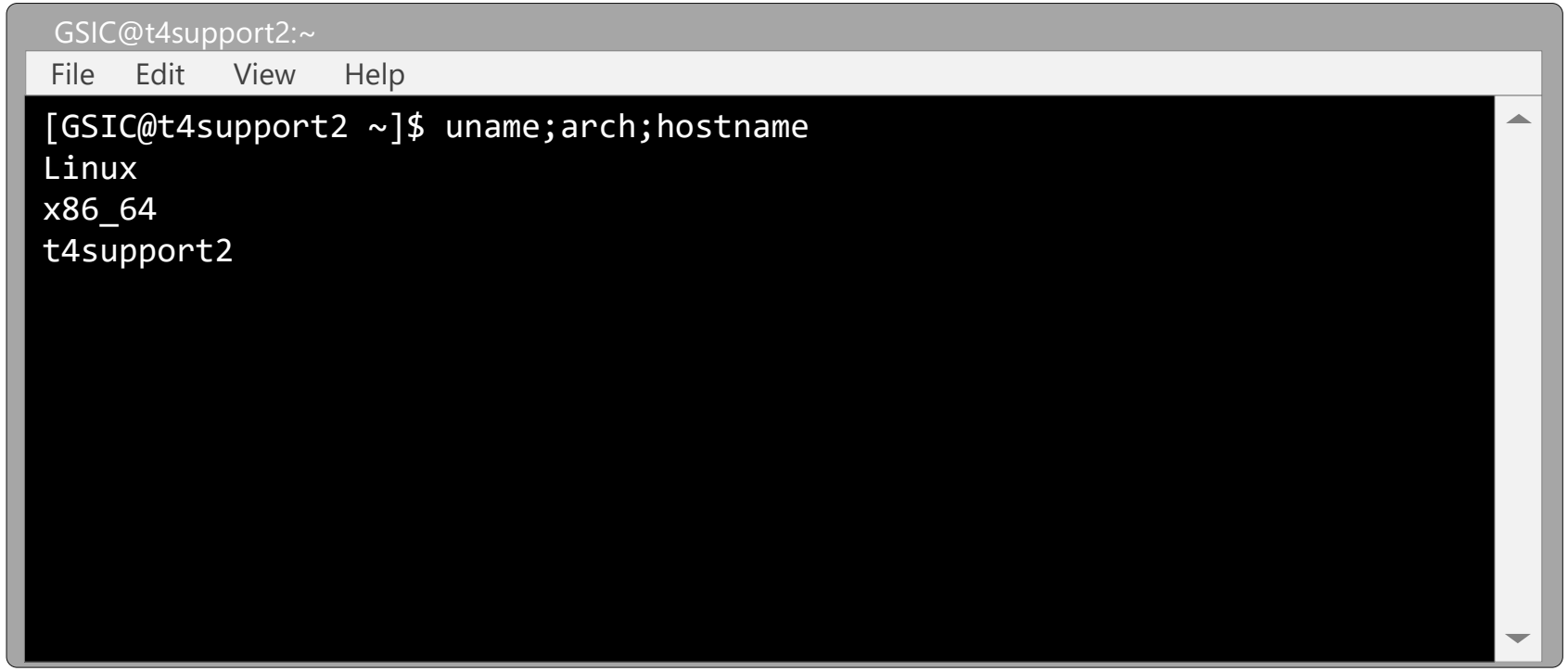
Command concatenation

Process can be executed collectively as program by combining commands.

- Connect command by semicolon (;)
- Connect command by pipe (|)
- Write the output of the command to file
- Input file in command
- Shell Programming (Shell script)

Semicolon

- Connects multiple commands by semicolon (grouping)
- After execution of command 1, command 2 and command 3 are executed in sequence.



Pipe

- | (vertical bar)

Example:

Transfer the standard output obtained with cat to grep

GSIC@t4support2:~

File Edit View Help

```
[GSIC@t4support2 ~]$ cat /usr/mpi/gcc/openmpi-1.10.4/include/mpi.h | grep INDEX
#define MPI_T_ERR_INVALID_INDEX       57
  MPI_COMBINER_INDEXED,
  MPI_COMBINER_HINDEXED_INTEGER,
  MPI_COMBINER_HINDEXED,
  MPI_COMBINER_INDEXED_BLOCK,
  MPI_COMBINER_HINDEXED_BLOCK
```

Redirection

- Output Redirection
- >　　overwrite a file
- >>　　Append to a file

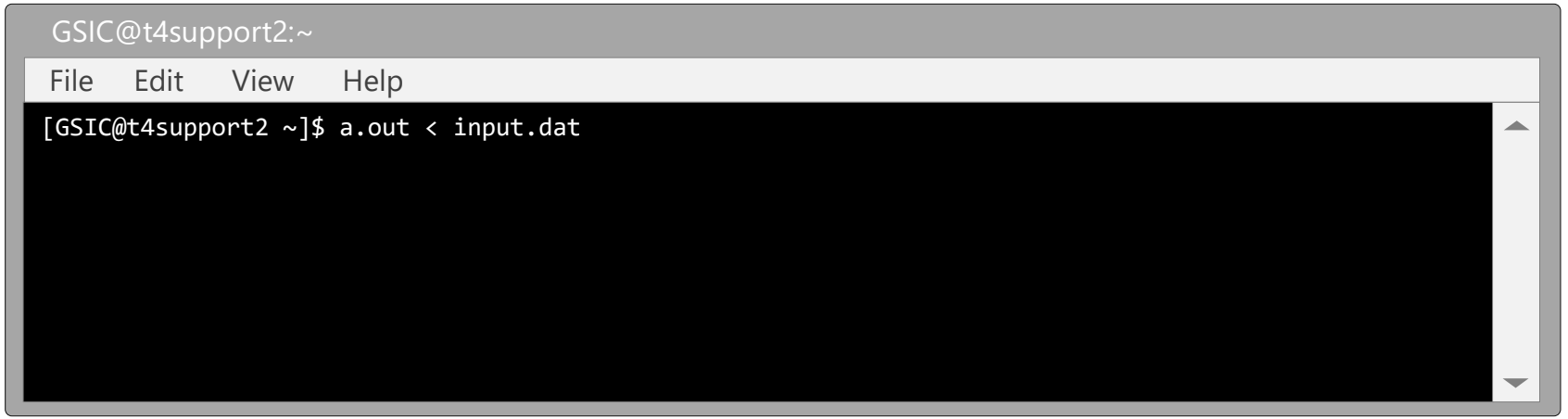
GSIC@t4support2:~

File　Edit　View　Help

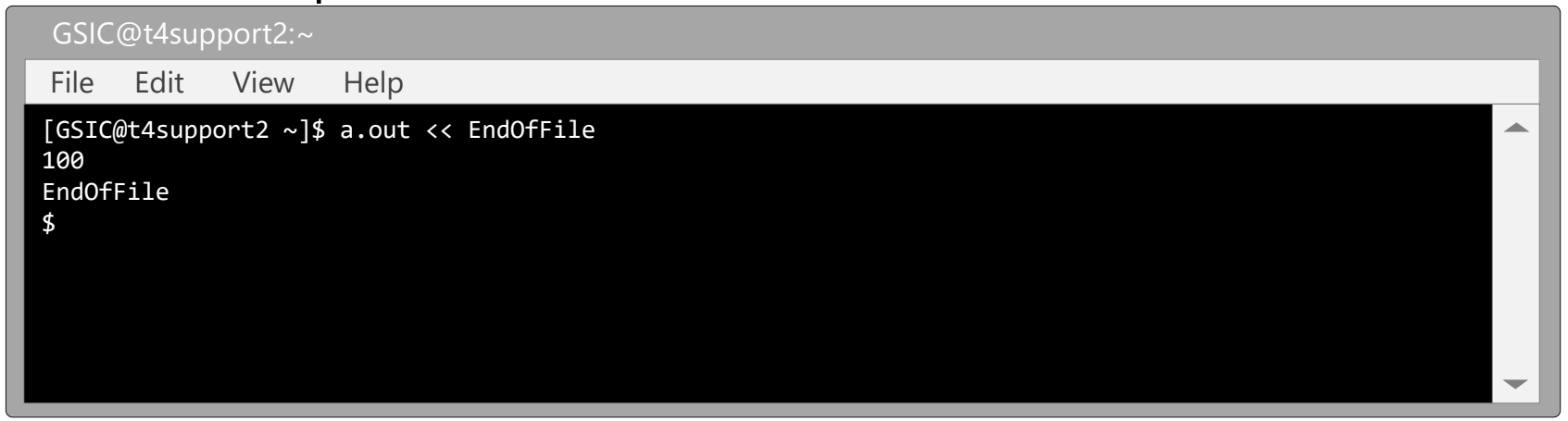
```
[GSIC@t4support2 ~]$ ls
Desktop hoge
[GSIC@t4support2 ~]$ ls > ls.txt
[GSIC@t4support2 ~]$ cat ls.txt
Desktop
hoge
[GSIC@t4support2 ~]$ ls >> ls.txt
[GSIC@t4support2 ~]$ cat ls.txt
Desktop
hoge
Desktop
hoge
ls.txt
```

Here document

- Transfer data from standard output to command (program)
  - Input from file



  - Input from standard input



Process

- Process Monitoring
  - ps : display running processes
  - top : monitor processes in real time
- Process is execution unit of processing on OS
  - The following example shows three processes running.

GSIC@t4support:~

File Edit View Help

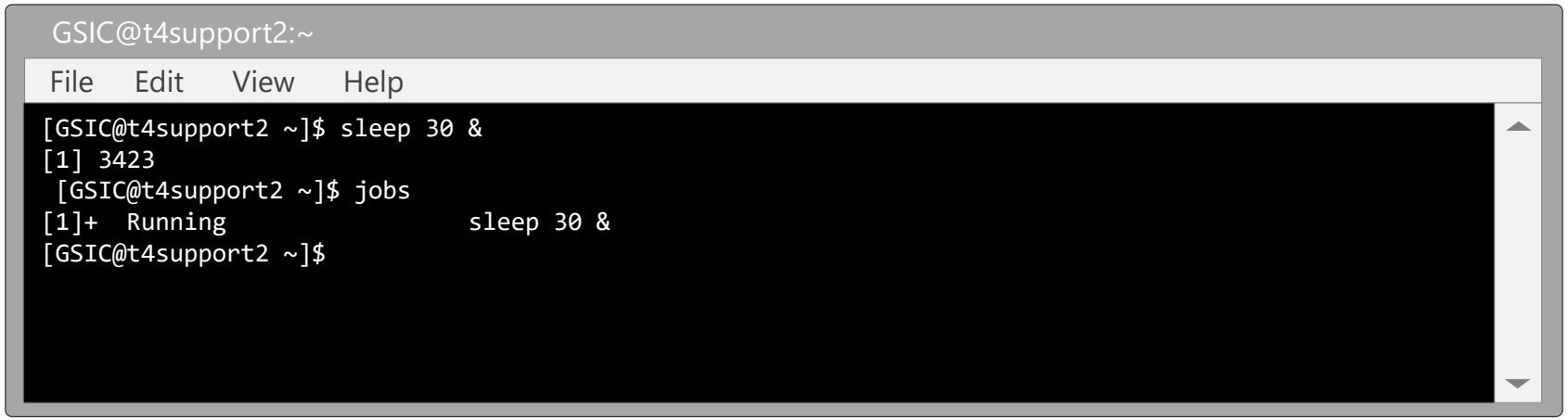
```
ux00000@r6n2:~> ps aux
USER               PID %CPU %MEM      VSZ    RSS   TTY  STAT  START   TIME COMMAND
ux00000         354797 13.2  0.1 52321684 490304 pts/1  Rl   13:19   0:02 pmemd.cuda.MPI
ux00000         354798 13.2  0.1 52321348 487148 pts/1  Rl   13:19   0:02 pmemd.cuda.MPI

ux00000@r6n2:~> top
  PID USER      PR  NI    VIRT    RES    SHR S   %CPU  %MEM     TIME+ COMMAND
354797 hpe_use+  20   0 49.897g 456836 408116 R 14.286 0.173   0:02.09 pmemd.cuda.MPI
354798 hpe_use+  20   0 49.897g 453680 407172 R 14.286 0.172   0:02.08 pmemd.cuda.MPI
```

To exit, type Q or Ctrl+C.

job

- A job is a shell execution unit that combines commands / programs
  - Even when a command is connected by a pipe or the like, it is called a job.
  - Use the jobs command to check running jobs.

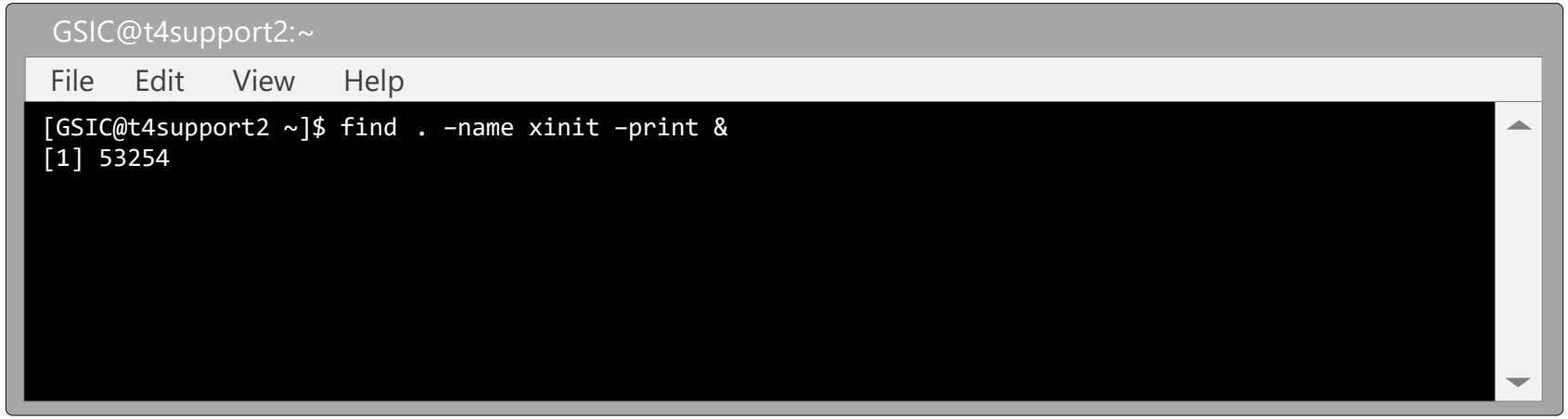
  - jobs : list jobs
  - fg : move a job to the foreground
  - bg : resume a job in the background



Background execution

- Commands (programs) that take a long time to execute
- To do another task on the same terminal after starting a program.
  → Execute the program as a background job

- To execute command, add “&” at the end of command.



Type "Ctrl+Z" and "bg" then current process in foreground will move to background as a job.

Terminating a Process

- Find the Process ID

  Use either ps or top command. (the example is shown below.)

GSIC@t4support:~

File Edit View Help

```
ux00000@r6n2:~> ps aux | grep ux00000
354797 13.2  0.1 52321684 490304 pts/1 Rl   13:19   0:02 pmemd.cuda.MPI -O -i input -p top -o test
354798 13.2  0.1 52321348 487148 pts/1 Rl   13:19   0:02 pmemd.cuda.MPI -O -i input -p top -o test

ux00000@r6n2:~> top
354797 hpe_use+  20   0 49.897g 456836 408116 R 14.286 0.173   0:02.09 pmemd.cuda.MPI
354798 hpe_use+  20   0 49.897g 453680 407172 R 14.286 0.172   0:02.08 pmemd.cuda.MPI
```

Process ID

- Execute kill command to stop the process

GSIC@t4support:~

File Edit View Help

```
ux00000@r6n2:~> kill 354797 354798

# Forcely
ux00000@r6n2:~> kill -9 354797 354798
```

Exercise

- Let's use the commands that have learned so far.
- Please execute the commands shown here in order.

① cd
② mkdir lesson
③ cd lesson
④ cp -r /gs/bs/soudan/UNIX/* ./
⑤ cp sample.sh sample.txt
⑥ file sample.txt
⑦ vi sample.txt
  - Please edit somewhere appropriately in a file with vi
  - after editing, type Esc
  - :wq (save and quit)
⑧ diff sample.sh sample.txt
⑨ sdiff sample.sh sample.txt

Run programs on compute node (Job Submission)

- module command to set environment
- Batch job scheduler
- How to submit job

module command to set environment

- When using TSUBAME4, load execution environment of the application you want with the module command. It needs to be done before executing the application.
- Example (Intel Compiler):

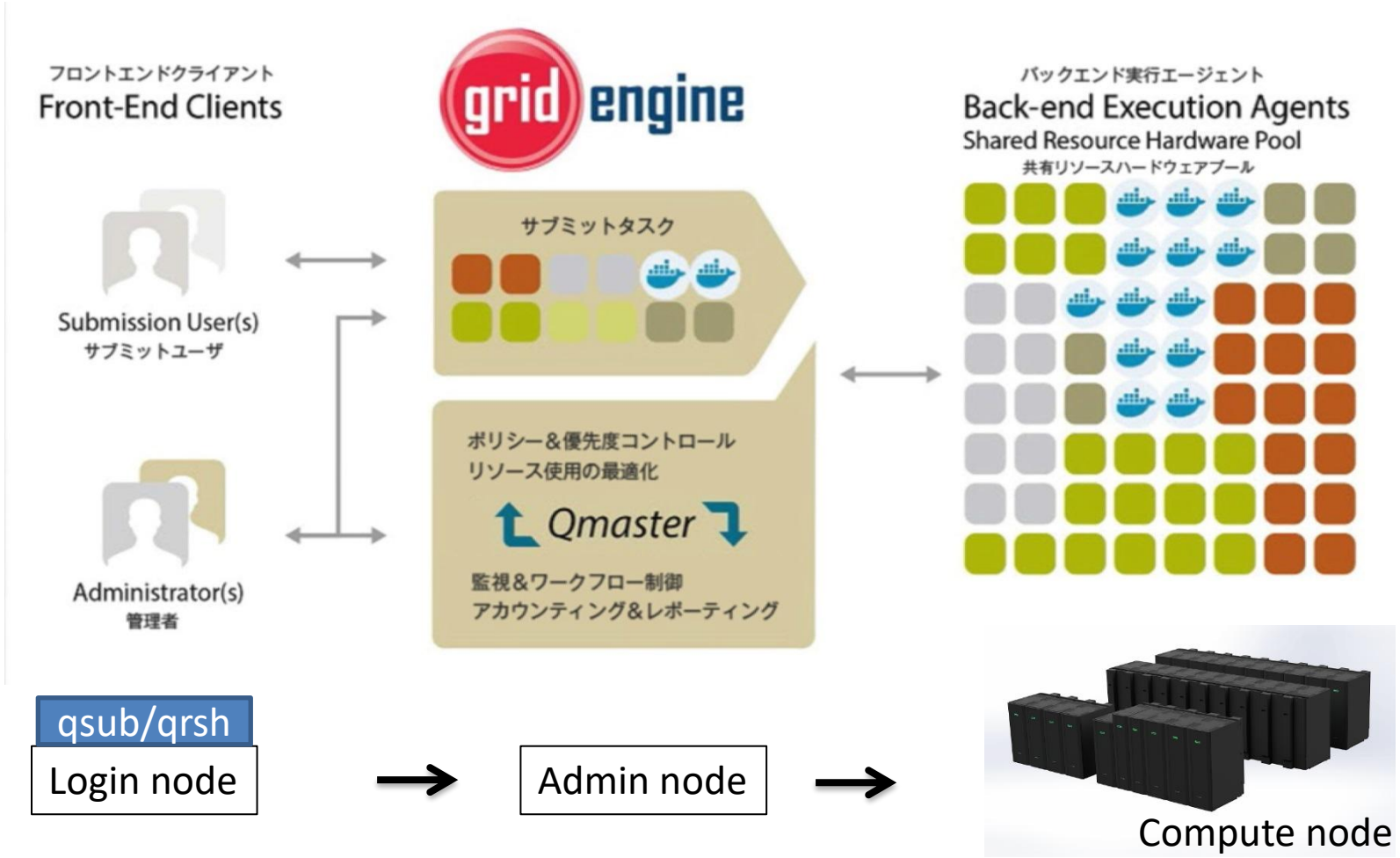
  $ module load intel
- To browse available modules

  $ module avail

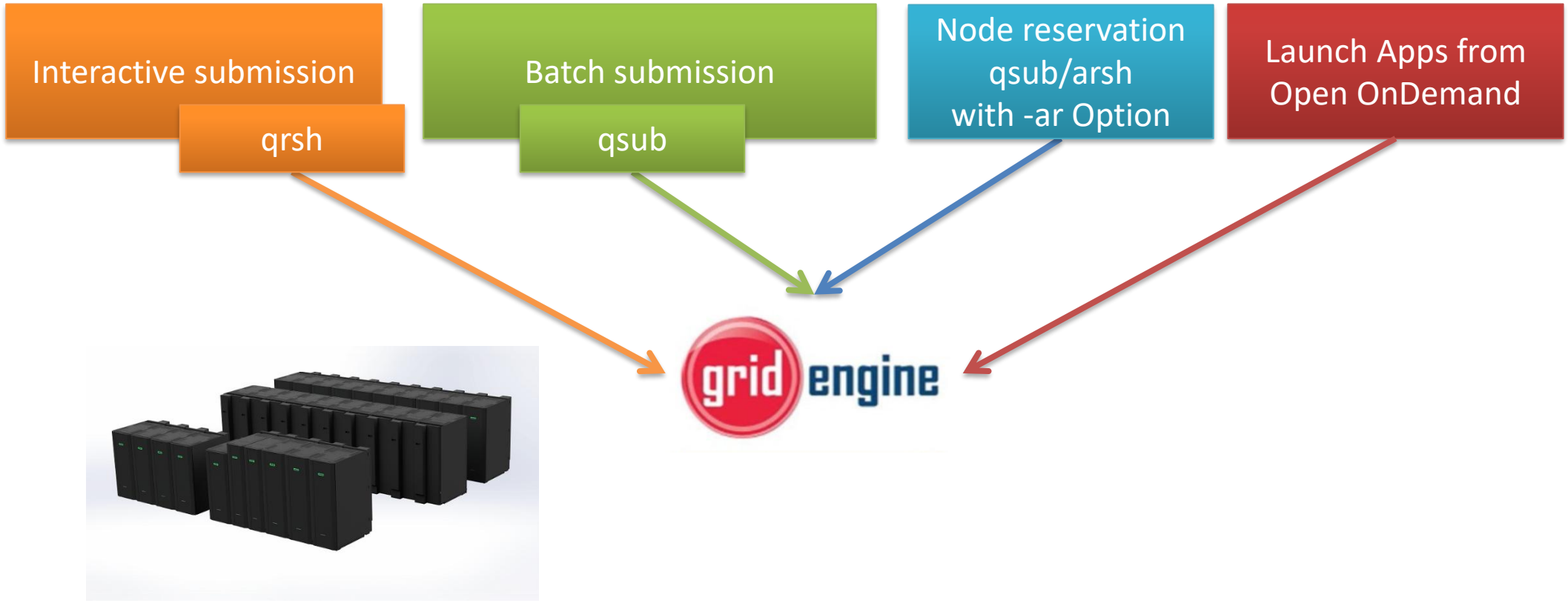
Command	Operation
module avail	List available modules
module load	Load specific module's environment
module list	List loaded modules
module purge	Purge loaded modules

Batch job scheduler

- Altair Grid Engine (AGE) is introduced as batch job scheduler.



Use Compute node (qsub/qrsh)



Flow from file creation to submit

1. Prepare program you would like to use.
2. Consider the resources required for the job. (num. nodes, run time length)
3. Select one of resource types.
   - node_f, node_q, cpu_4, or ...
4. Create a batch job script for job scheduler.
5. Submit the job by qsub.
6. Check the job status and logs to confirm successful execution.
7. Check the result of the program after the job finished.

Resource Types

- Only node_f is permitted to login by SSH from another terminal.
- Effective use of TSUBAME point, appropriate

Type	CPU cores	Memory (GB)	GPUs	Local scratch area (GB)
node_f	192	768	4	1920
node_h	96	384	2	960
node_q	48	192	1	480
node_o	24	96	1/2	240
gpu_1	8	96	1	240
gpu_h	4	48	1/2	120
cpu_160	160	368	0	96
cpu_80	80	184	0	48
cpu_40	40	92	0	24
cpu_16	16	36.8	0	9.6
cpu_8	8	18.4	0	4.8
cpu_4	4	9.2	0	2.4

Limits of job submission

- Limits
  - Run time
    - Maximum running time is 24 hours per job. (Reservation execution might be able to run for 1 week in some cases)
  - Number of available resources at the same time
    - 6144 slots (12288 slots on weekend)
  - Maximum degree of parallelism per job
    - 64 (Note: maximum effective capacity of node_f will be 32, because of the 6144-slot limitation)
  - Number of running jobs at the same time
    - 30 jobs per user (100 jobs on weekend)

If you submit jobs exceeding the limitations, the exceeded jobs' status become wait and does not run until the preceding running job ends.

*slots = physical cpu cores

See Resource Limit Values for more info.

Run programs on compute node

When a Job is submitted, The job scheduler will randomly allocate One/multiple node(s) of 240 compute node to a Job. Users cannot select a specific compute node.

The followings are the commands to submit jobs.

- Interactive job submission (for small to medium scale)
  - Perform *qrsh* command. It is for directory log in to an interactive node.

    $ qrsh -g *TSUBAME_group* -l *resource_type* -l *max_run_time*
  - Then you can run programs on the node.

    $ ./a.out
- Batch job submission (for medium to large scale)
  - Perform *qsub* command. It is for submitting a job, on a login node.

    $ qsub -g *TSUBAME_group* -l *resource_type* -l *max_run_time batch_script.sh*

  For using node reservation, add *-ar reservation_number* to qsub argument.

Submit job（Create job script)

- Example of a job script

Refer to TSUBAME4.0 User's Guide.
https://www.t4.cii.isct.ac.jp/docs/handbook.en/jobs/#jobscript

```
#!/bin/bash              <-shebang
#$ -cwd                  <-set to run on current directory
#$ -N test_job           <- job name
#$ -l cpu_4=1           <- resource type
#$ -l h_rt=0:3:0        <- max. run time

echo "this host is" `hostname` "."
```

Submit job（batch job）

- qsub -l *resource_type* -l max_run_*time* -g *TSUBAME_group jobscript.sh*

See also https://www.t4.cii.isct.ac.jp/docs/handbook.en/jobs/#execute_qsub

GSIC@t4support:~

File Edit View Help

```
[ux00000@login1 ~]$ qsub -g TSUBAME_group sample.sh

#For trial, submit a job without -g option. Note that the limitation to a
job execution is within 3 min and 2 nodes.
[ux00000@login1 ~]$ qsub sample.sh
```

Submit Job (interactive job)

- qrsh -l *resource_type* -l *reserve_time* -g *TSUBAME_group*

Refer to TSUBAME4.0 User's Guide.
https://www.t4.cii.isct.ac.jp/docs/handbook.en/jobs/#interactive

GSIC@t4support2:~

File Edit View Help

```
#General usage
#Interactive execution using node_f as resource type
[ux00000@login1 ~]$ qrsh -g GSIC -l node_f=1 -l h_rt=8:0:0

#For trial, submit a job without -g option. Note that the limitation to a
job execution is within 3 min and 2 nodes.
[GSIC@login1 ~]$ qrsh -l node_f=2 -l h_rt=0:3:0
[GSIC@r5i6n5 ~]$ (Run command here.)
```

batch queue control

- Confirm that the job(s) has been submitted with *qstat*.

GSIC@t4support2:~

File Edit View Help

```
GSIC@login1:~> qstat
job-ID     prior   name       user         state submit/start at     queue    jclass slots ja-task-ID
-----------------------------------------------------------------------------------------------------
93501      0.55500 PDF        GSICUSER00   r    10/04/2017 07:39:58 all.q@r2i4n6    56
93578      0.55500 QRLOGIN    GSICUSER00   r    10/04/2017 11:39:58 all.q@r6i3n2    28
```

- Delete the running job with *qdel*

GSIC@t4support2:~

File Edit View Help

```
GSIC@login1:~> qdel 93578
GSIC@login1:~> qstat

GSIC@login1:~>
```

User environment

- Shell
- Environment variables
- HISTSIZE shell variable
- PATH variable
- File transfer
- SSH
- X-window system

Shell

- A shell is a program that provides access to operating system services to a human user or other programs.
- The major compatible shells which TSUBAME4 supports, which available with chsh command, are as follows:
  - /bin/bash
  - /bin/tcsh
  - /bin/zsh
- *chsh* command to change a current shell
  - $ chsh /bin/tcsh

  It takes 5 minutes to reflect the change.

Shell Environment

In case of bash

- The order of profiles (configuration files) read when log in

  /etc/profile /etc/bashrc

  ~/.bash_profile

  ~/.bash_login (if ~/.bash_profile is not exist)

- ~/.bashrc is loaded whenever bash is executed.
- Editing the personal configuration file ~/.bashrc
  - PATH="/usr/local/bin:$PATH“
  - export PATH
- After editing, re-login or execute “source .bashrc” to reflect.
- It is preferable to write alias here.
- In case of editing files, check properly before updating.
- Check it in another terminal.

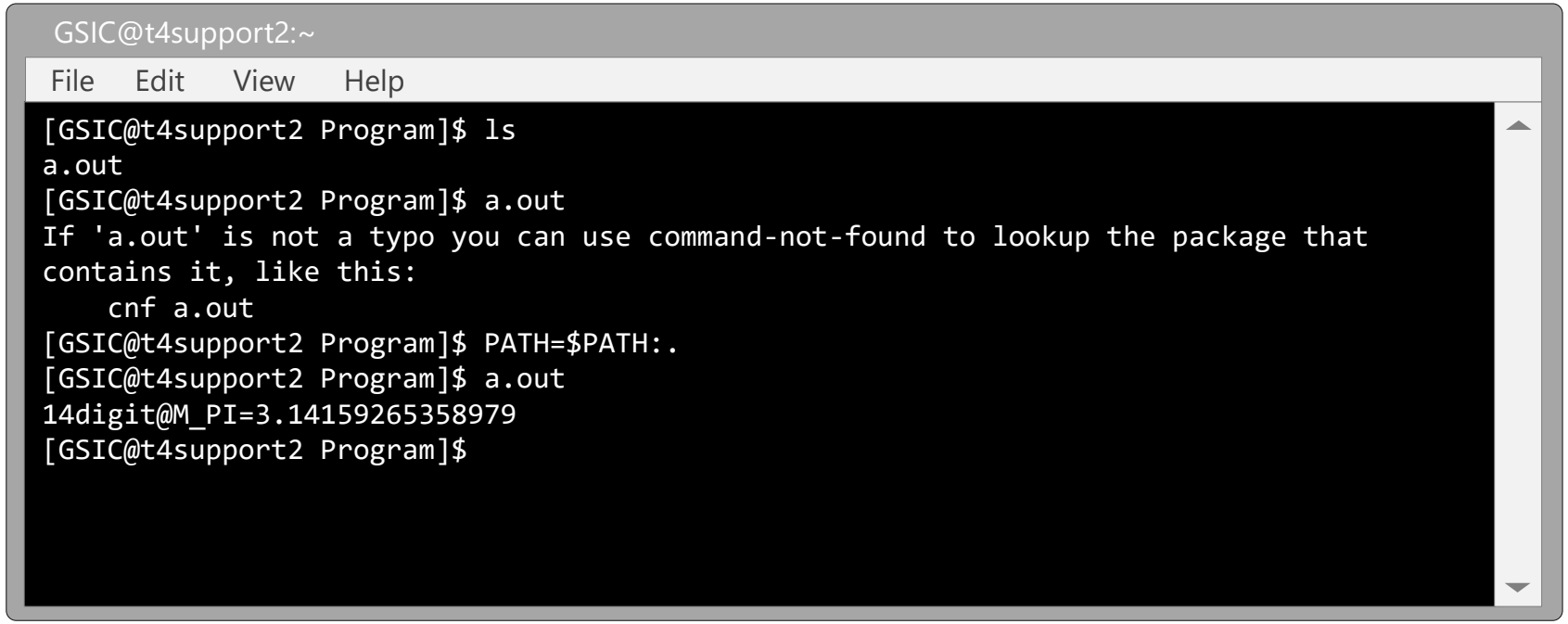
HISTSIZE shell variable

- History
  - Stores a predetermined number of commands most recently.
  - It can be useful when executing the same (or similar) command.
  - The number of history to be stored can be set arbitrarily.
- Examples
  - $ export HISTSIZE=600 ⇒ History size will be set 600
  - $ history 3 ⇒ Shows 3 items in history list

```
534 cd
535 ls
536 history 3
```

PATH variable

- Set the location of command.
- The setting sometimes needs to be modified. For example, when a command or a program are installed or created by an individual.



※ If it is described in .bashrc, it will be always enabled.

File transfer

- A system connected via a network is called a host.
- It is easy to transfer files between hosts via TCP/IP.
- Commands for the function
  - ftp, rcp, rsync, sftp, scp
- rsync, sftp, scp are available in TSUBAME.

Examples of rsync/sftp/scp

GSIC@t4support:~

File Edit View Help

```
#rsync
[GSIC@t4support ~]$ rsync -av --progress -e "ssh -i .ssh/id_ecdsa -l ux00000"
login.t4.gsic.titech.ac.jp:/gs/bs/soudan/UNIX/testfile ./
receiving incremental file list
testfile
         990 100%  966.80kB/s    0:00:00 (xfer#1, to-check=0/1)

sent 42 bytes  received 1078 bytes  2240.00 bytes/sec
total size is 990  speedup is 0.88
[GSIC@t4support ~]$

#sftp
[GSIC@t4support ~]$ sftp -i ~/.ssh/id_ecdsa ux00000@login.t4.gsic.titech.ac.jp
Connected to login.t4.gsic.titech.ac.jp.
sftp> get /gs/bs/soudan/UNIX/testfile
Fetching /gs/bs/soudan/UNIX/testfile to testfile
/gs/hbs/soudan/UNIX/testfile
sftp> exit

#scp
[GSIC@t4support ~]$ scp -i ~/.ssh/id_ecdsa
ux00000@login.t4.gsic.titech.ac.jp:/gs/bs/soudan/UNIX/testfile .
testfile
```

X Window system

- Used in various operating systems including Linux.
- Used for GUI applications
- Hardware independent
  - The system is pre-installed in Linux/Mac
  - Lots of software for Windows
    - Cygwin
    - MobaXterm
    - PuTTY/Tera Term/RLogin+VcXsrv/Xming

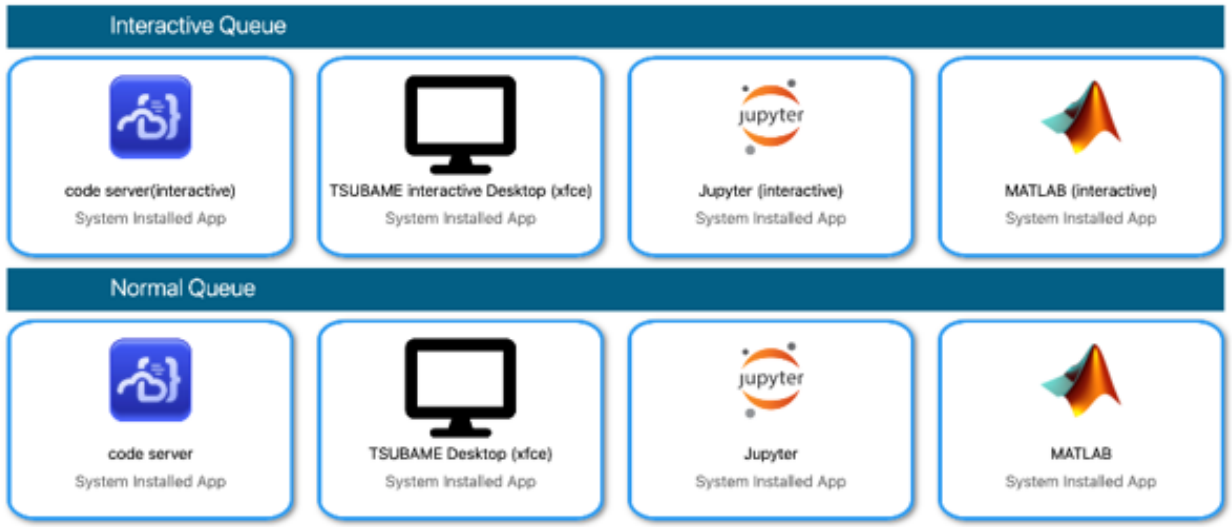
X Window system

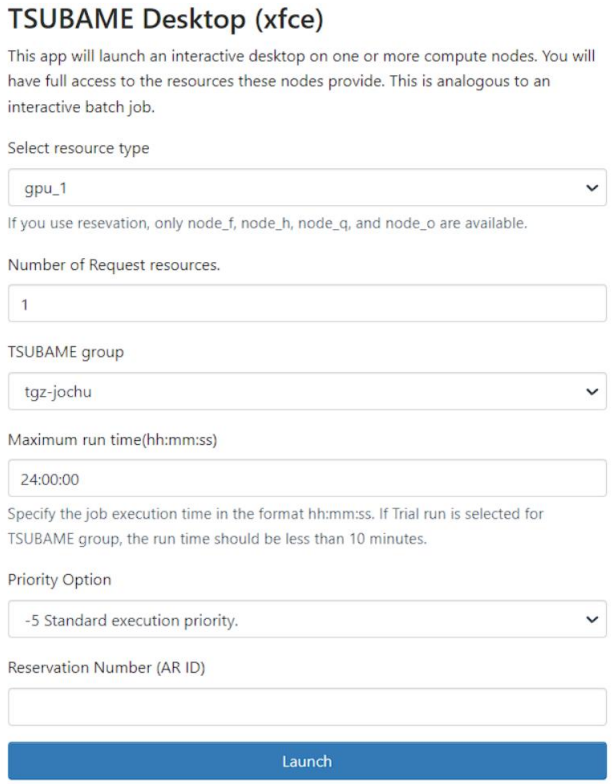
- Running X Window applications
  → Client program, server program
- X protocol, unique communication procedure
  → Data exchange between client/server
    - Server side (Displayed side)
      Definition to allow display request from system
      % xhost +*client_host_name*
    - Client side (Displaying side)
      Specifies DISPLAY environment variables.
      % export DISPLAY=*server_host_name*

Please use Open OnDemand to run GUI application.

Usage of compute node (Open OnDemand)

- https://ood.t4.gsic.titech.ac.jp/
- Applications described as "interactive" are for free within campus.
- To use this service, check the followings:
  - Already configured password for your TSUBAME account
  - The e-mail sent from this service will be received.





for details, https://www.t4.cii.isct.ac.jp/docs/ood/ (In Japanese)

Conclusion

That's all !

Thank you!

Get more information about T4

- Manuals
  - https://www.t4.cii.isct.ac.jp/en/manuals
  - TSUBAME4.0 User's Guide
    - https://www.t4.cii.isct.ac.jp/docs/handbook.en/
  - TSUBAME Portal User's Guide
    - https://www.t4.cii.isct.ac.jp/docs/portal.en/
  - Migration from TSUBAME3.0 to TSUBAME4.0
    - https://www.t4.cii.isct.ac.jp/docs/migration.en/

- This document is available at https://www.t4.cii.isct.ac.jp/en/lectures