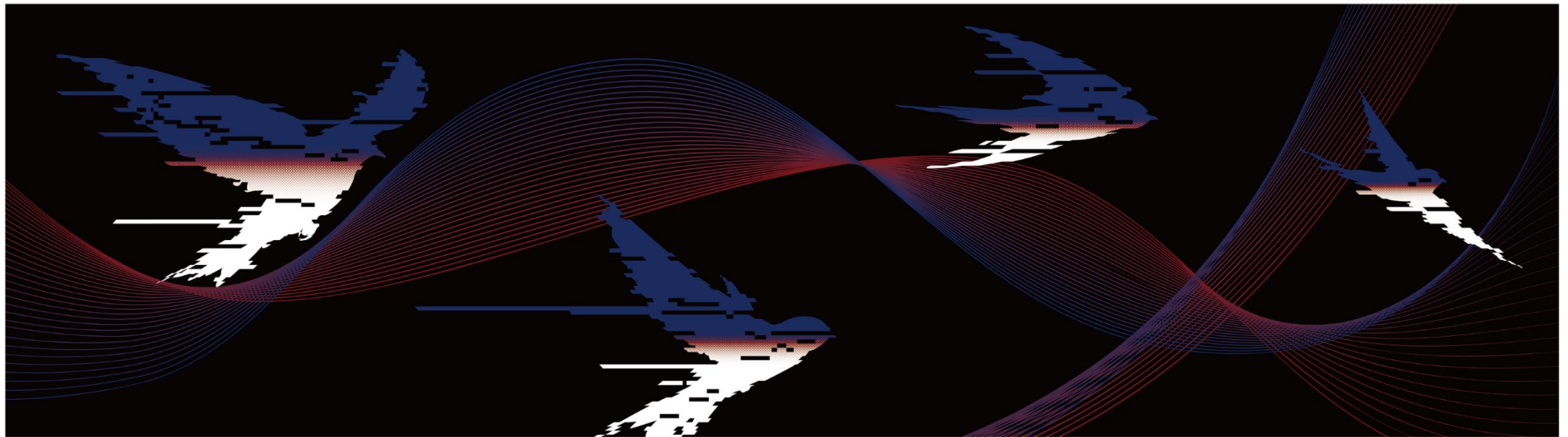# Introduction to TSUBAME
(Linux basics)

Contents

- Overview of TSUBAME4
- Introduction of Linux
- File operation
- Various commands
- Run programs on compute node (Job Submission)
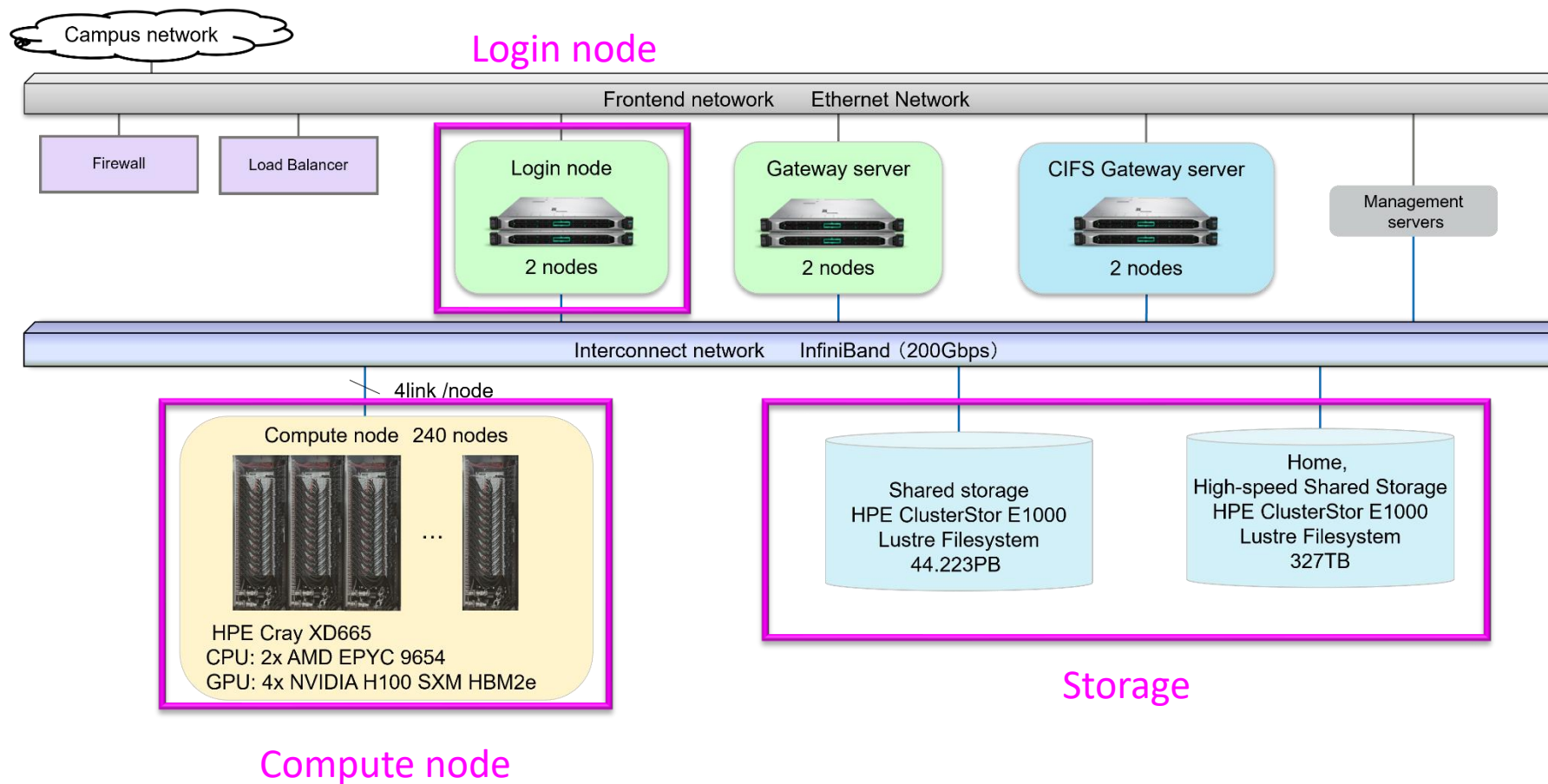- User environment

Overview of TSUBAME4

- System details
- Commercial applications
- Provided services
- Getting Started
- Usage of compute node
- T4 web page introduction

TSUBAME4

- TSUBAME4 is the Super Computer introduced in on April 1st, 2024.

- Total 960 GPUs (NVIDIA H100)
- Theoretical computing performance
  - double precision performance (FP64) : 66.8 PFLOPS
  - half precision performance (FP32) : 952 PFLOPS
- Total memory capacity : 180 TiB
- Total SSD Capacity: 327TB
- Total HDD Capacity: 44.2PB
- 200 Gbps High speed network (inside T4 system)

- As a new utility, users can use computing resoures through Web browser (Open OnDemand).
  - Various resource selection from 4 cpu cores to high paralell computing / multiple GPUs for per job.
  - Obviously, the standard method (SSH) is provided.

- Programs developed at TSUBAME3 can be used without any changes.
  - Combination of x86_64 CPU x NVIDIA GPU x Linux OS

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

T4 System architecture

Campus network

Login node

Frontend netowork   Ethernet Network

| Firewall | Load Balancer | **Login node** 2 nodes | Gateway server 2 nodes | CIFS Gateway server 2 nodes | Management servers |

Interconnect network   InfiniBand（200Gbps）

4link /node

Compute node 240 nodes

…

HPE Cray XD665
CPU: 2x AMD EPYC 9654
GPU: 4x NVIDIA H100 SXM HBM2e

Compute node

Shared storage
HPE ClusterStor E1000
Lustre Filesystem
44.223PB

Home,
High-speed Shared Storage
HPE ClusterStor E1000
Lustre Filesystem
327TB

Storage

System details - Compute node

- HPE Cray XD665　240 nodes

Item	Specification
CPU	AMD EPYC 9654 (2.4GHz) × 2CPU
# of cores/threads	96 cores / 192 threads × 2CPU
Memory size	768GiB
GPU	NVIDIA H100 SXM5 94GB HBM2e × 4
SSD	1.92TB NVMe U.2 SSD
Interconnect	InfiniBand NDR200 200Gbps × 4

System details - Storage and SW

- Storage

Areas	Mount point	Capacity	Filesystem
High-speed storage area Home directory (SSD)	/gs/fs /home	372TB	Lustre
Large-scale storage area Common application deployment (HDD)	/gs/bs /apps	44.2PB	Lustre
Local scratch area	/local	1.92TB/node	xfs（SSD）

- Software
  - OS:Red Hat Enterprise Linux Server 9.3
  - Commercial applications (see the following 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

Activation is required for each application before use.

Activation can be done on T4 portal with TSUBAME point.

In a job session, environment for each application will be configured by module command.

Commercial applications except for Gaussian, debugger and compiler are for users on-campus.

Provided services

- ## Compute node
  - TSUBAME4 provids 240 nodes. (HPE Cray XD665)

- ## Storage
  - home directory (up to 25GiB for each user) : for free
    - home directory for each user is located in /home/[0-9]/*username.*
    - Accessible from all nodes on the system.
  - High-speed/Largh-scale storage area : paid service (TSUBAME point required)
    - Group disk area composed of Lustre file system
    - High-speed storage -> SSD, Large-scale storage -> HDD
  - Scratch area (SSD) : for free (available in running job)
    - Local scratch area
    - Shared scratch area

Paid Services in TSUBAME4

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

- When you activate paid services, "TSUBAME point" is required.
- You will get amount of "TSUBAME point" with your budget of research project.
- TSUBAME point will be expired until the end of faculty year. Cannot carry over to the next year.

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

1. Get an TSUBAME account (See https://www.t4.gsic.titech.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 subadmin.]
   - Add users to the group [by group admin. and users]
   - Grant permission to users [by group admin. or subadmin.]
   - Configure group disks [by group admin. or auth. users]
   - Apply for using Apps. (activation) [by group admin. or auth. users]
   - <Other configuration if you need>
5. Submit jobs

Setup at TSUBAME4 potal

Usage of compute node

- There are mainly 2 types of using compute node.

1. CLI use: SSH
2. Web based use: Open OnDemand <-- New

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

Usage of compute node (SSH)

- Login to login node (SSH key authentication)
  - Perform the ssh command to access to *login.t4.gsic.titech.ac.jp* in a terminal, on your PC, then automatically login to one of the login node by load balancing.

※On login node, running heavy processes are prohibited. When monitoring system finds such a process, it will kill them automatically and forcely.

Integrated Development Environment (IDE) such as VS Code also uses large amount of resorces.

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

User's local PC		Login node		Compute node
Terminal	→ SSH →	login.t4.gsic.titech.ac.jp login1 or login2 (Load balancing)	→ Submit job →	Compute node
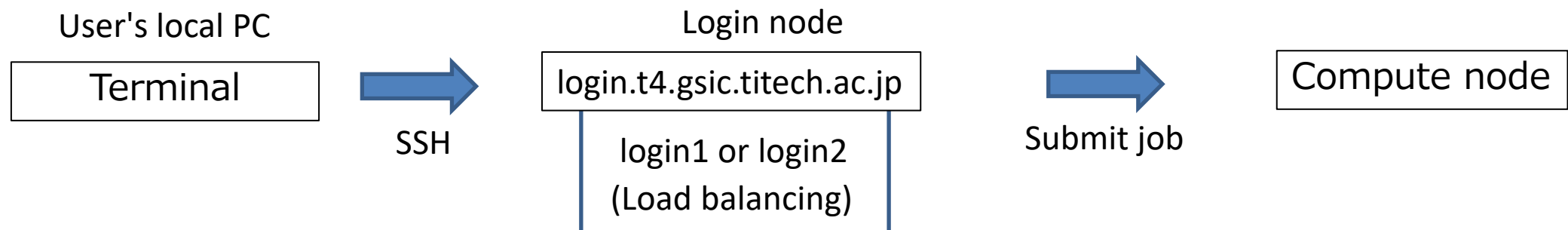
TSUBAME information

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

- TSUBAME4 portal
  - https://portal.t4.gsic.titech.ac.jp/
    For user/group setting (e.g. SSH pubkey registration, point parchase, node reservation)
- X (formaly known as Twitter)
  - @TSUBAME_sc
    The latest news will be announced on X and T4 Homepage.

- If there is some inquiriy about TSUBAME, post at "Contact Us" on T4 Homepage
  - https://www.t4.gsic.titech.ac.jp/en/contact-t4

Introduction of Linux

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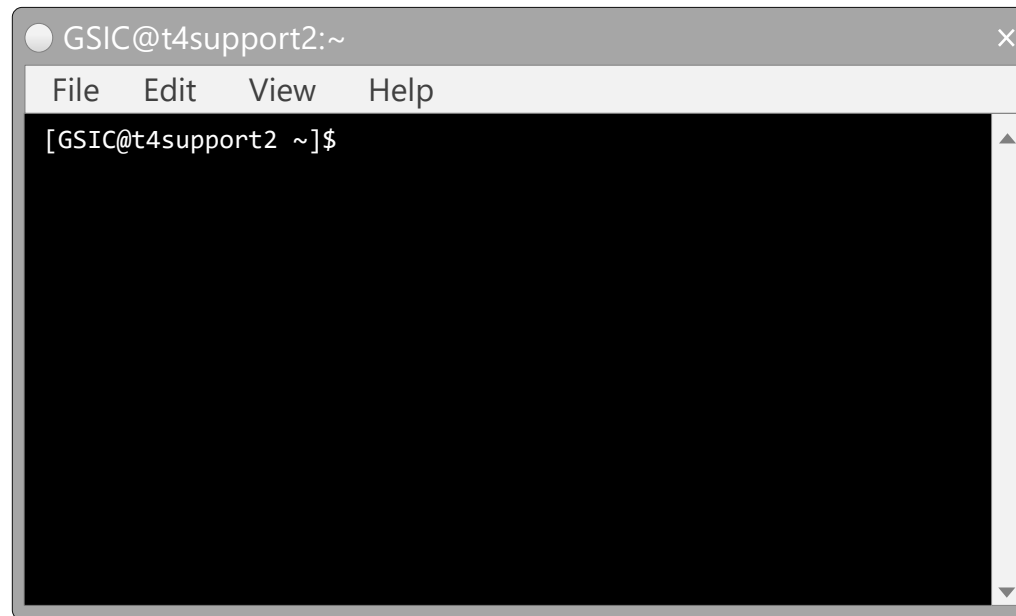
Overview of Linux

- 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
  - Slackware Linux
  - SUSE Linux Enterprise Server (SLES)
  - Red Hat Enterprise Linux (RHEL)
  - CentOS
  ...
- Authentication
  - Local login
    - username / password
  - Remote login
    - username / password
    - Key authentication <-- TSUBAME4 supports this authentication.

Terminal

- Using a standard terminal on Linux.
- The line which has a symbol such as "%", "$" and ">" is called prompt. (In the example below, $ is displayed.)
- Type a command on this line.

```
GSIC@t4support2:~                                              ×
File    Edit    View    Help
[GSIC@t4support2 ~]$
```

Terminal emulators for Windows

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

- MobaXterm seems to be convenient because they contains several services including X window system, VNC, port forwarding, ...

- Anyway, please try several terminal emulators and select the one you like.

See also Available SSH client on Windows

Use PowerShell

- Start -> Type "powershell" in the search box and Enter.
- Execute SSH command.

(Make sure to configure SSH key pair in advance)
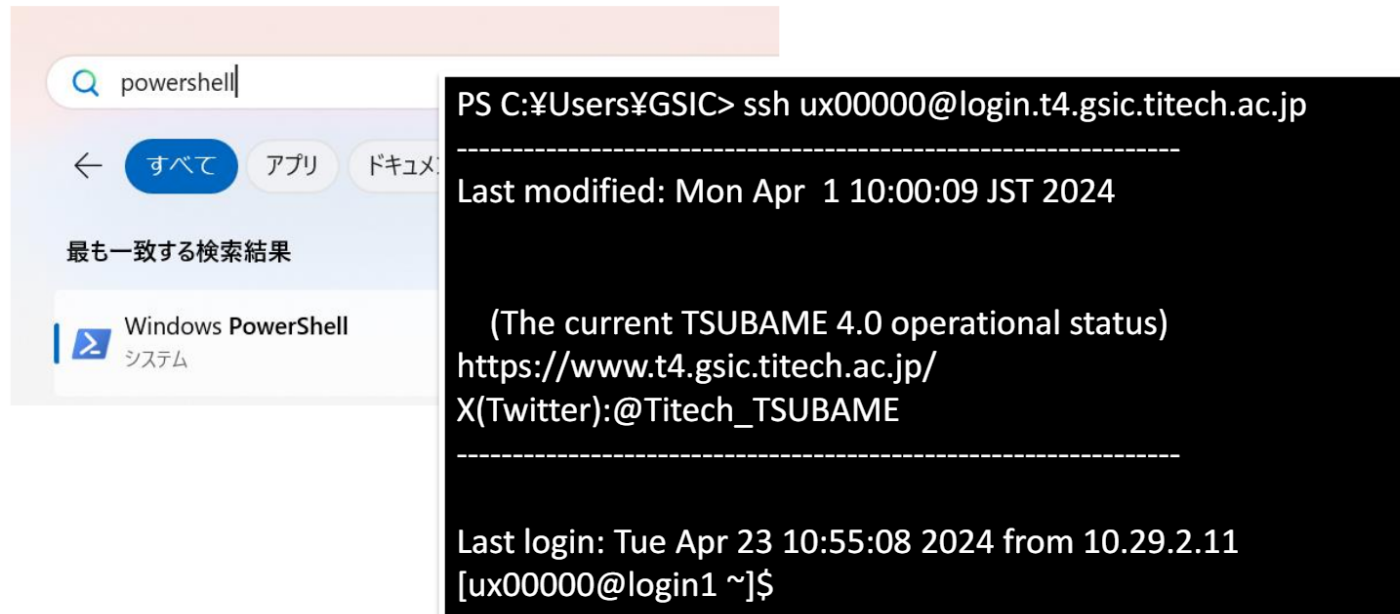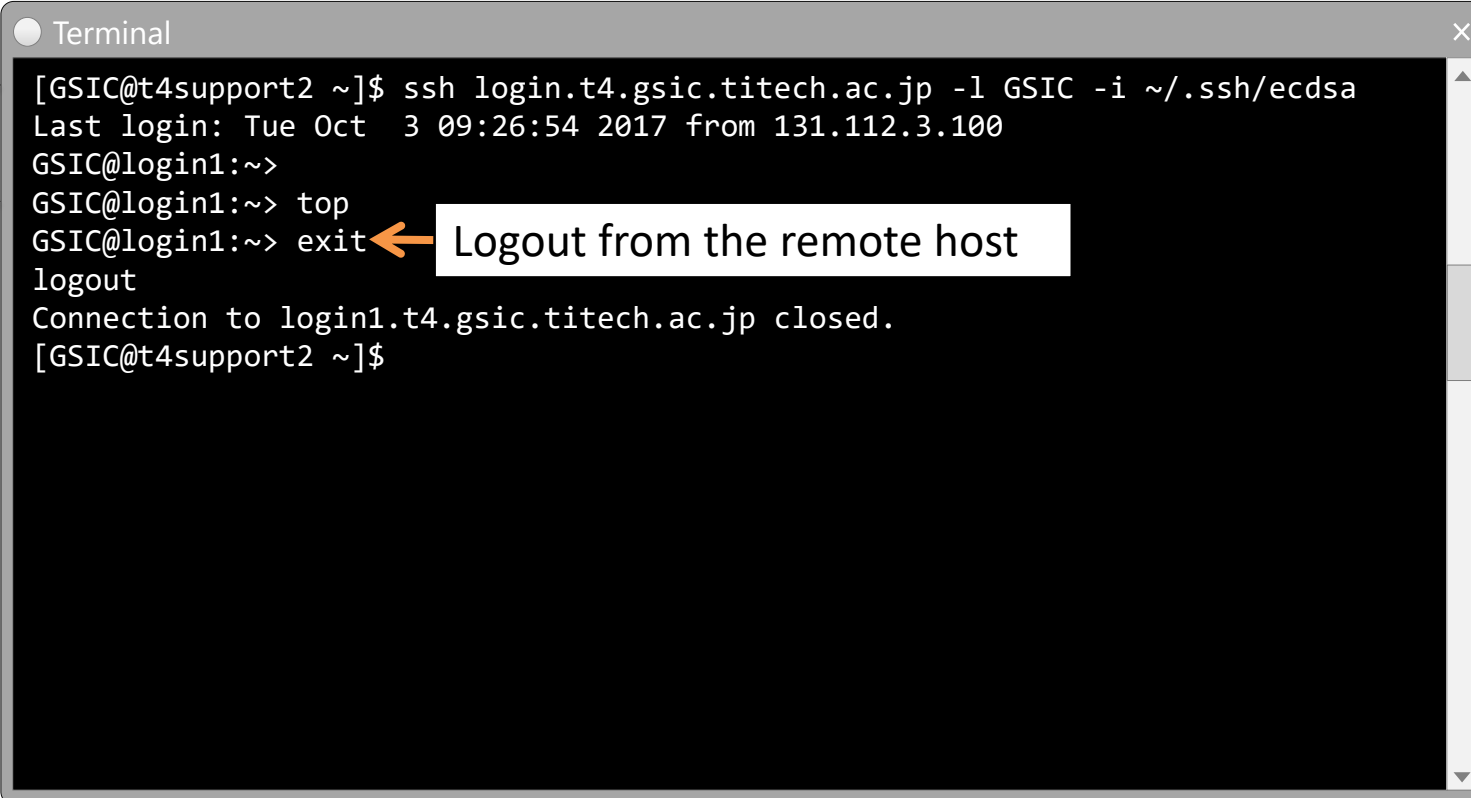
Use terminal on Mac

- Applications >Utilities >Terminal.app
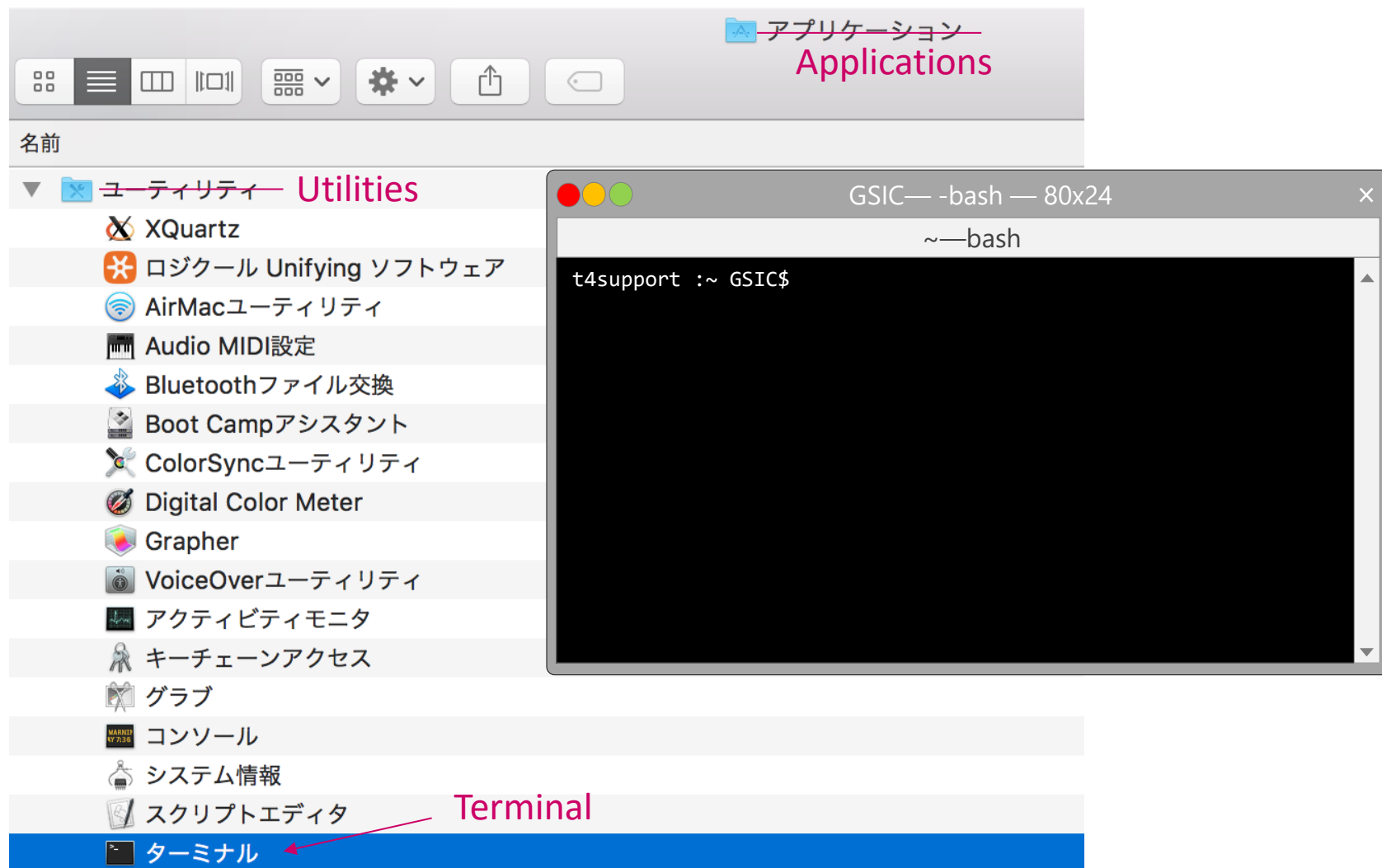
Remote login

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

Authentication →

Remote operation →

```
[GSIC@t4support2 ~]$ ssh login.t4.gsic.titech.ac.jp -l GSIC -i ~/.ssh/ecdsa
Last login: Tue Oct  3 09:26:54 2017 from 131.112.3.100
GSIC@login1:~>
GSIC@login1:~> top
GSIC@login1:~> exit
logout
Connection to login1.t4.gsic.titech.ac.jp closed.
[GSIC@t4support2 ~]$
```

Logout from the remote host

Terminal

SSH key authentication

The authentication system consisits of a pair of keys, called public key and private key.

- Advantage
  - Only accessible from the machine which have the key = High security
- Disadvantage
  - It's a little hard for configure and management.

[NOTE] Recommend to set passphrase
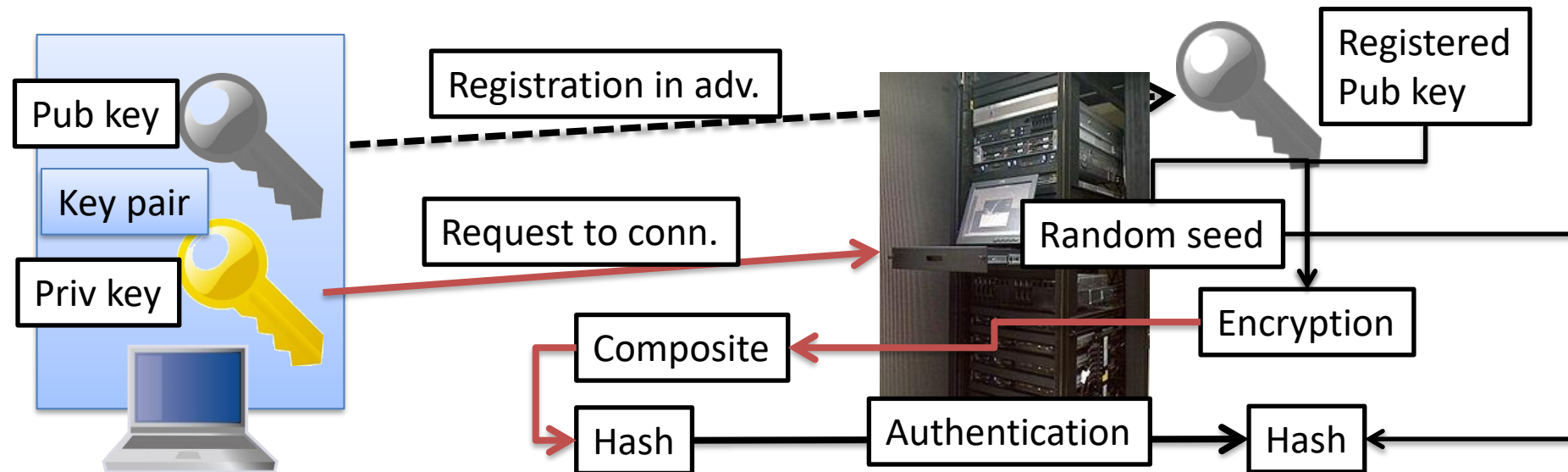
Creating SSH key pair

- Use ssh-keygen command

  See https://www.t4.gsic.titech.ac.jp/docs/faq.en/general/#keypair

```
GSIC— -bash — 80x24
~—bash
t4support :~ GSIC$ ssh-keygen -t ecdsa
```

Upload public key

Upload public key to TSUBAME via T4 poral.

• https://portal.t4.gsic.titech.ac.jp/ptl/user/sshPublicKey

Check the text of public key (.pub), copy and paste it to the T4 potal page.
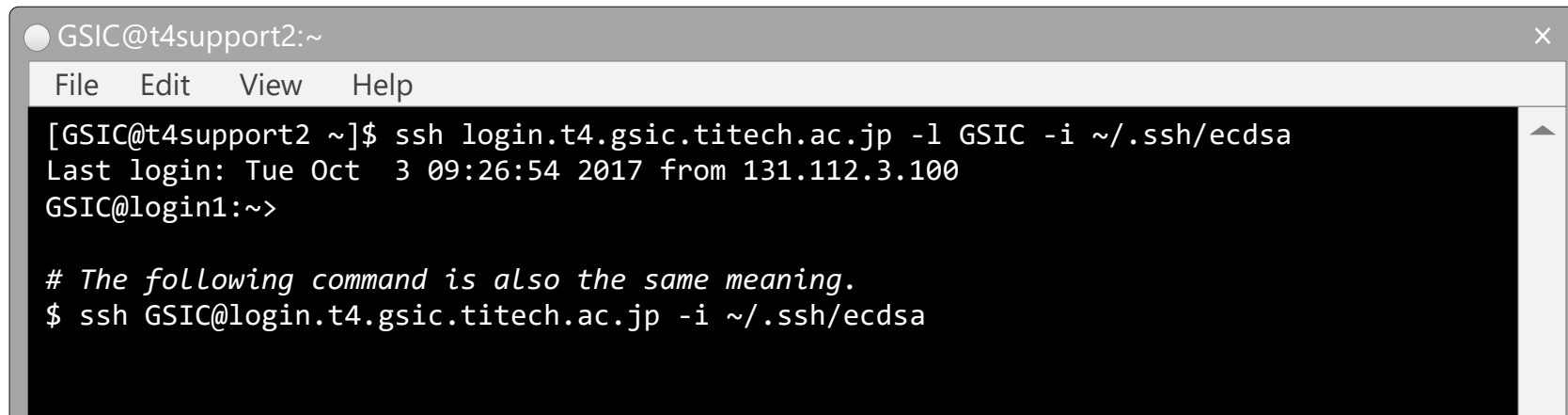
```
GSIC— -bash — 80x24
~—bash

t4support :~ GSIC$ cat ~/.ssh/id_ecdsa.pub
ssh-ecdsa
AAAAgtnvguirnfguiesvbguigrfguih345895r4huj9oienvgbrwfr3h
2-fnvgaeonvgmewdg90tjug0nwvg0rjhrvfjedrf2 GSIC@t4support
```

Exercise (Login to TSUBAME4)

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

```
GSIC@t4support2:~                                              ×

File    Edit    View    Help

[GSIC@t4support2 ~]$ ssh login.t4.gsic.titech.ac.jp -l GSIC -i ~/.ssh/ecdsa
Last login: Tue Oct  3 09:26:54 2017 from 131.112.3.100
GSIC@login1:~>

# The following command is also the same meaning.
$ ssh GSIC@login.t4.gsic.titech.ac.jp -i ~/.ssh/ecdsa
```

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.

```
GSIC@t4support:~

File    Edit    View    Help

[GSIC@t4support ~]$ cat ~/.ssh/config
        ServerAliveInterval 120
        ServerAliveCountMax 30
```

File Operation

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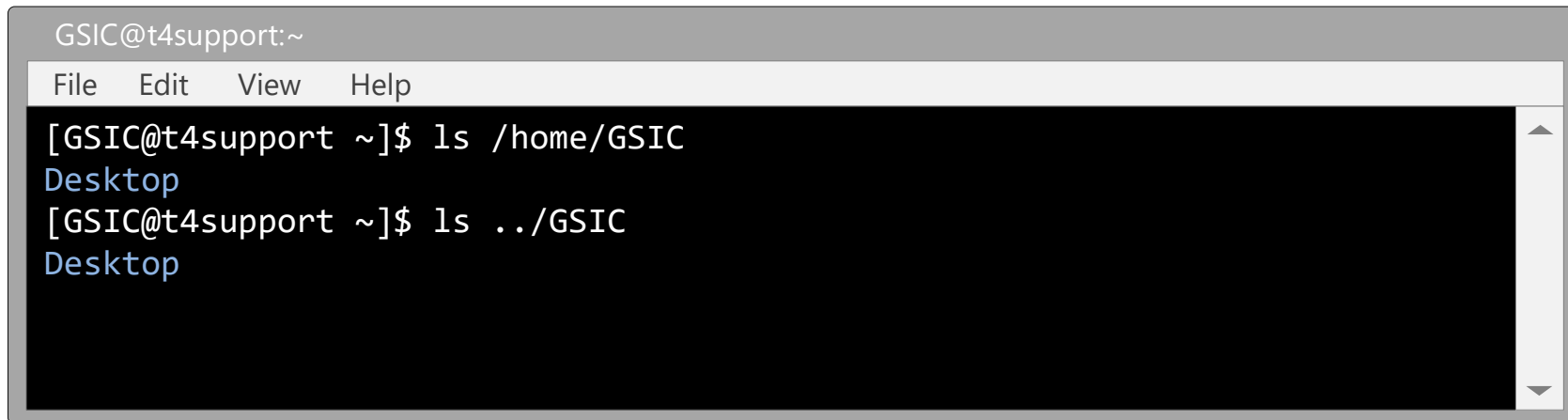
Filesystem

- Represented in hierarchical structure
- File management with directory
- Ordinary file and special file
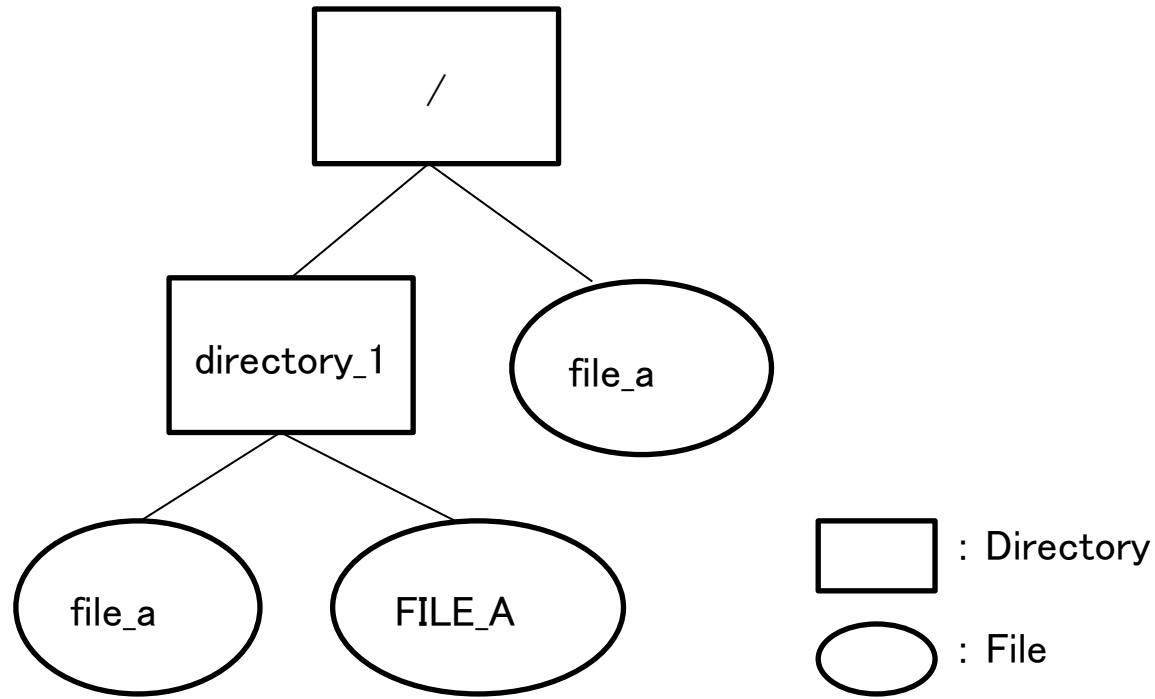
File specification

- The top of a hierarchical structure of a file system is called "root", represented as "/".

- How to access files
  - Absolute specification: Full specification from the root (/)
  - Relative specification: Specifies the location from current location

```
GSIC@t4support:~

File    Edit    View    Help

[GSIC@t4support ~]$ ls /home/GSIC
Desktop
[GSIC@t4support ~]$ ls ../GSIC
Desktop
```

Symbols

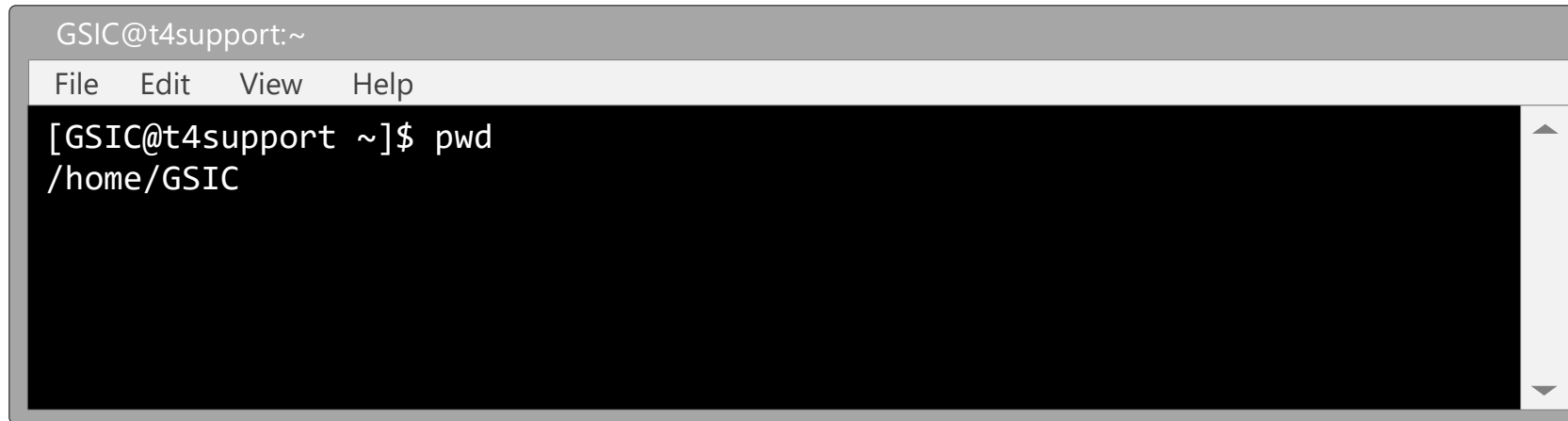- Command to confirm directory: pwd
- Home directory: ~
- One directory up: ..
- Current directory: .

```
GSIC@t4support:~

File    Edit    View    Help

[GSIC@t4support2 ~]$ pwd
/home/GSIC
[GSIC@t4support2 ~]$ cd ..
[GSIC@t4support2 home]$ cd ~
[GSIC@t4support2 ~]$
```

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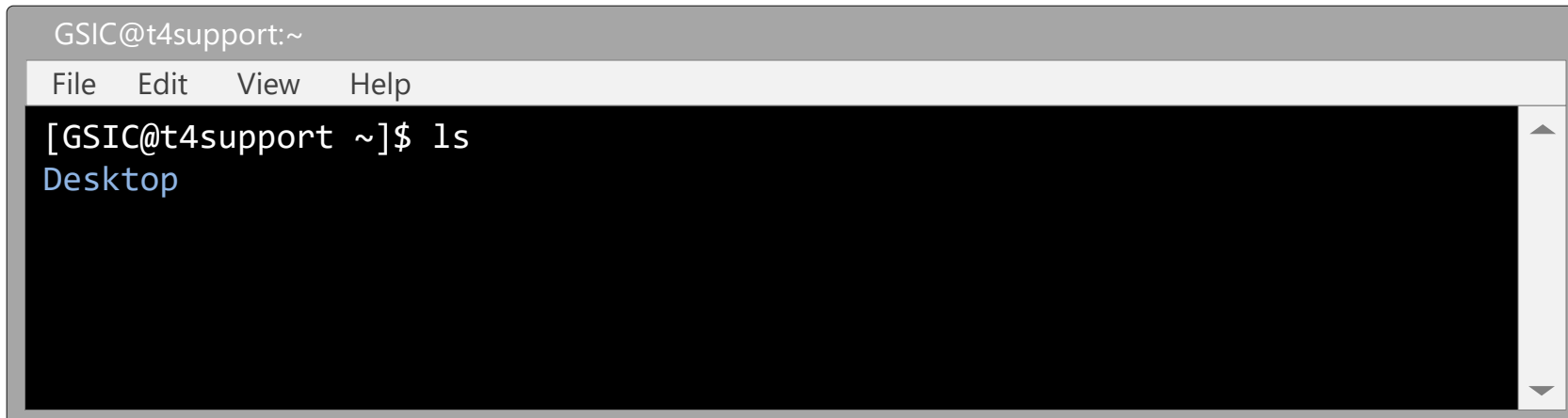
Information display commands

- pwd (to confirm where you are)

```
GSIC@t4support:~
File    Edit    View    Help
[GSIC@t4support ~]$ pwd
/home/GSIC
```

- ls (to display files in current or specified directory)

```
GSIC@t4support:~
File    Edit    View    Help
[GSIC@t4support ~]$ ls
Desktop
```

Special characters

- Special characters representation 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 any meaning itself, 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 |

- Mode

drwxrwxr-x

File type
d: directory
l: link
-: ordinary file

Owner's permission

Group's permission

Other's permission

Permission
r: readable
w: writable
x: executable
-: unauthorized
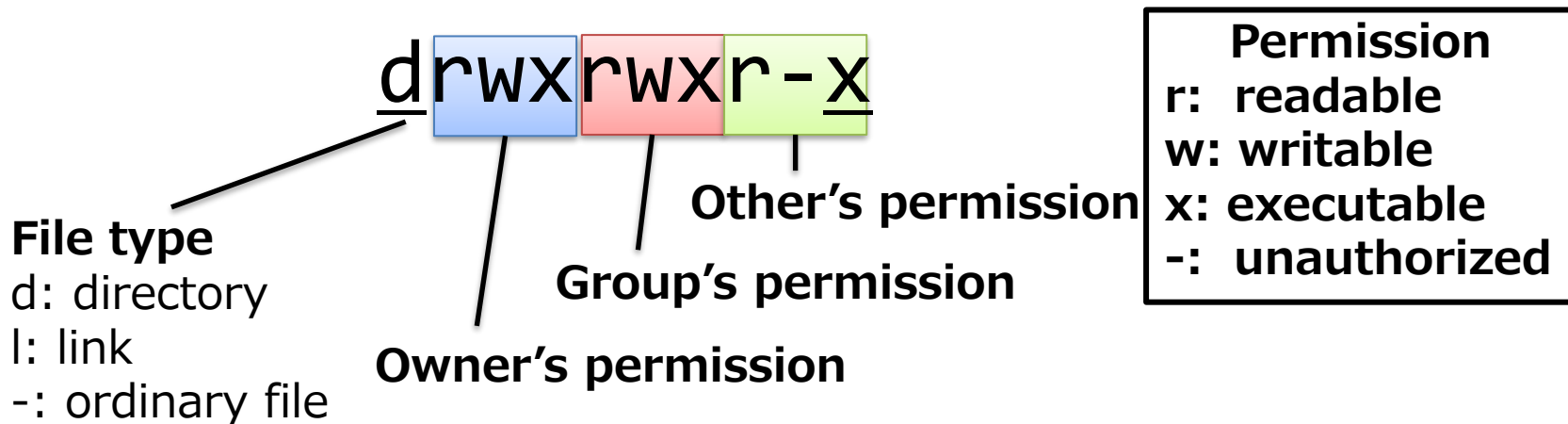
Commands for file management

- Create directory   % mkdir aaa

- Remove directory   % rmdir aaa

- Change file attribute   % chmod 755 aaa

       *755 is called as bit representation, and it represents access rights for owner, group and others.
       0 --- unauthorized
       1 --x execute only
       2 -w- write only
       3 -wx
       4 r-- read only
       5 r-x
       6 rw-
       7 rwx full access
       (Operation of administrator is as follows)

- Change owner   # chown necapps aaa

- Change group   # chgrp procon aaa

Different codes depending on OS

- Newline difference in text file
  - Unix/linux   ¥n LF(line feed)
  - Macintosh   ¥r CR(carriage return)
  - Win/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
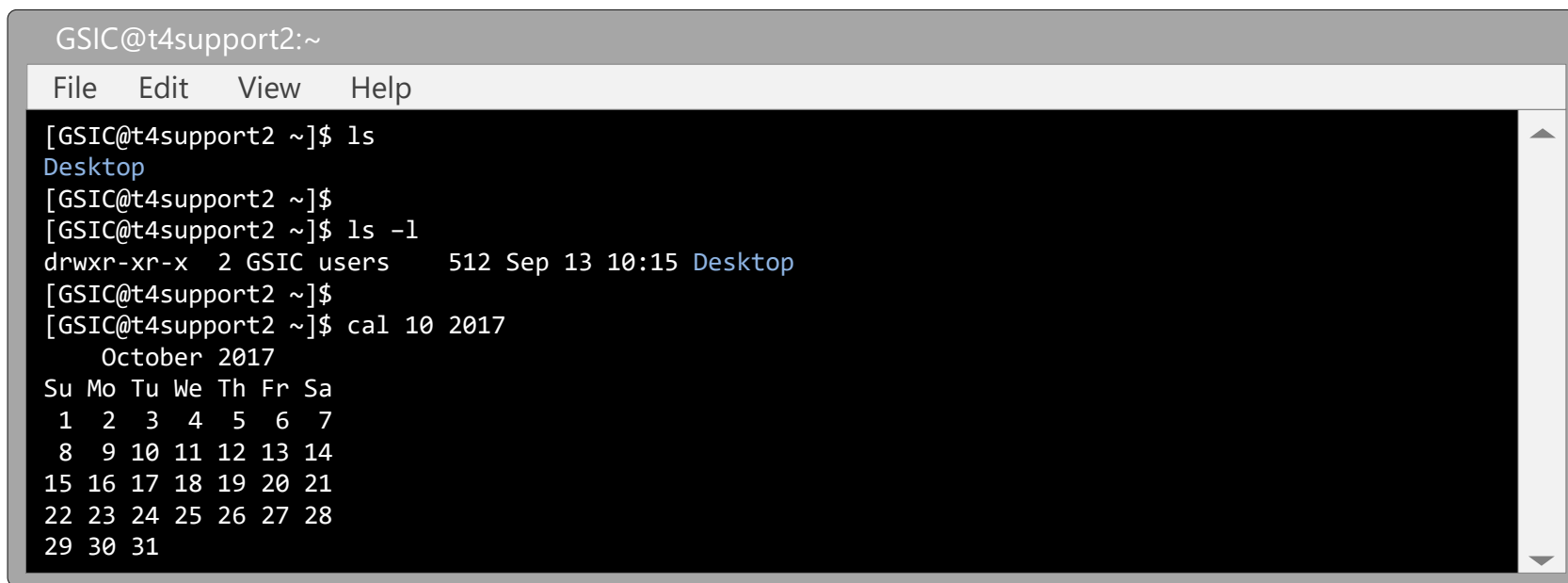- Online manual
- Command concatenation

- 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 operation

Usage

- Execute a command without any option
- Execute a command with options or arguments
- It is possible to combine multiple commands.
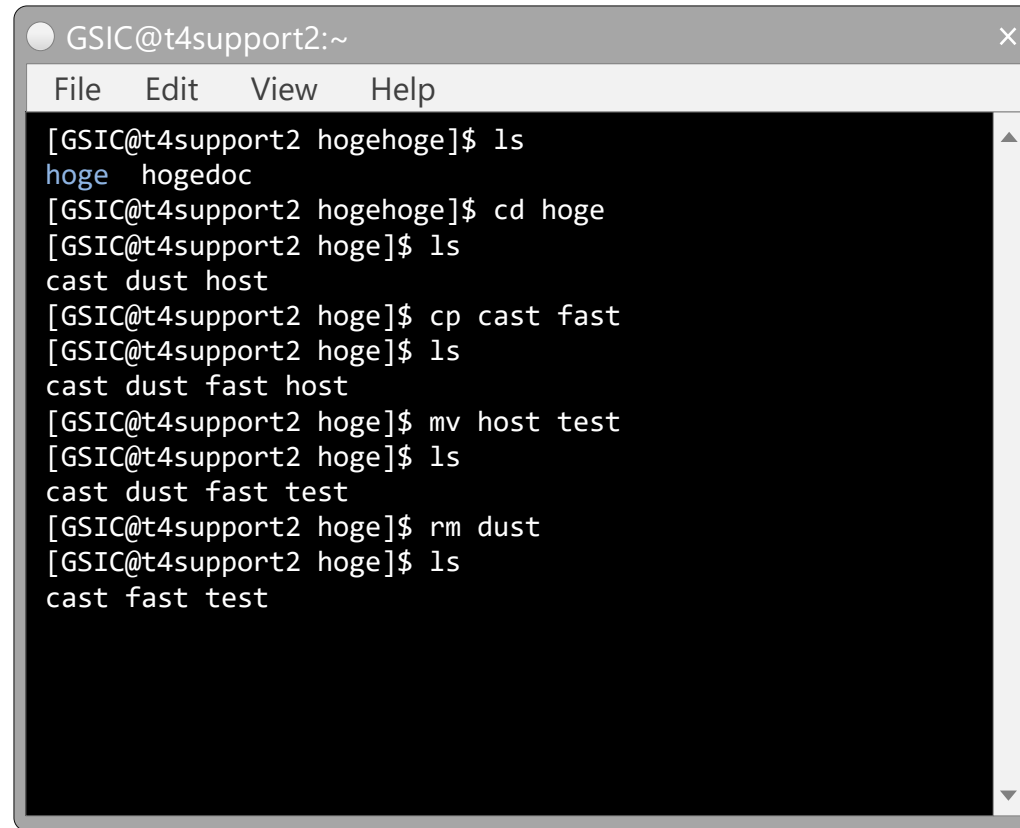
```
GSIC@t4support2:~

File    Edit    View    Help

[GSIC@t4support2 ~]$ ls
Desktop
[GSIC@t4support2 ~]$
[GSIC@t4support2 ~]$ ls -l
drwxr-xr-x  2 GSIC users     512 Sep 13 10:15 Desktop
[GSIC@t4support2 ~]$
[GSIC@t4support2 ~]$ cal 10 2017
     October 2017
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

Commands for file operation

- cd

  change directory

- cp

  copy

  copy a file or a directory

- mv

  move

  move a file or a directory

- rm

  remove

  remove a file or a directory

```
● GSIC@t4support2:~                          ×

  File    Edit    View    Help

[GSIC@t4support2 hogehoge]$ ls
hoge    hogedoc
[GSIC@t4support2 hogehoge]$ cd hoge
[GSIC@t4support2 hoge]$ ls
cast dust host
[GSIC@t4support2 hoge]$ cp cast fast
[GSIC@t4support2 hoge]$ ls
cast dust fast host
[GSIC@t4support2 hoge]$ mv host test
[GSIC@t4support2 hoge]$ ls
cast dust fast test
[GSIC@t4support2 hoge]$ rm dust
[GSIC@t4support2 hoge]$ ls
cast fast test
```

*In cp, mv, rm commands, users are prompted for confirmation by –i option.
*There is no command to restore the files that were removed once.

Alias function

- Adding alias to frequently used command or command line.
- alias can also be used as command.
- To unset alias, use unalias command.

```
GSIC@t4support2:~

File    Edit    View    Help

[GSIC@t4support2 ~]$ cat .bashrc
# rm option
alias rm='rm -i'
```
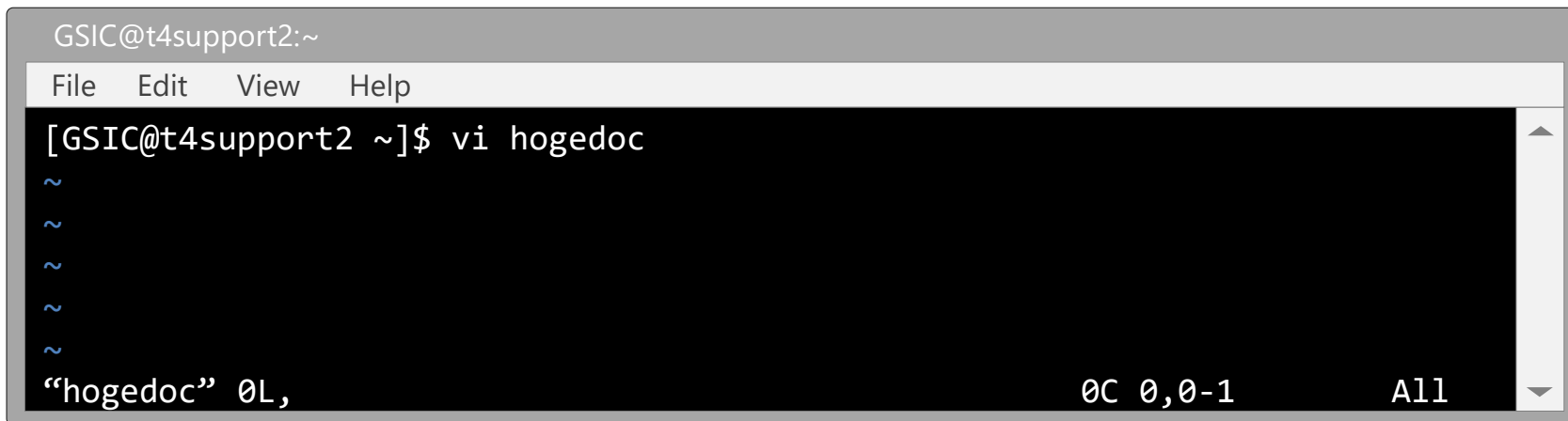
By writing this way in .bashrc, loss of file by mistake can be prevented.
By mistake, if space is given such as rm * .txt , then all files in current directry will be deleted.
By setting the alias, users are prompted for confirmation before removing by -i option.

Text editor (vi)

An editor necessarily installed in Linux machine

```
GSIC@t4support2:~

File    Edit    View    Help

[GSIC@t4support2 ~]$ vi hogedoc
~
~
~
~
~
"hogedoc" 0L,                                    0C 0,0-1        All
```

Insert mode by i or a

Standby mode by Esc key

Delete one character by x, delete one line by dd

Save and exit editor by :wq
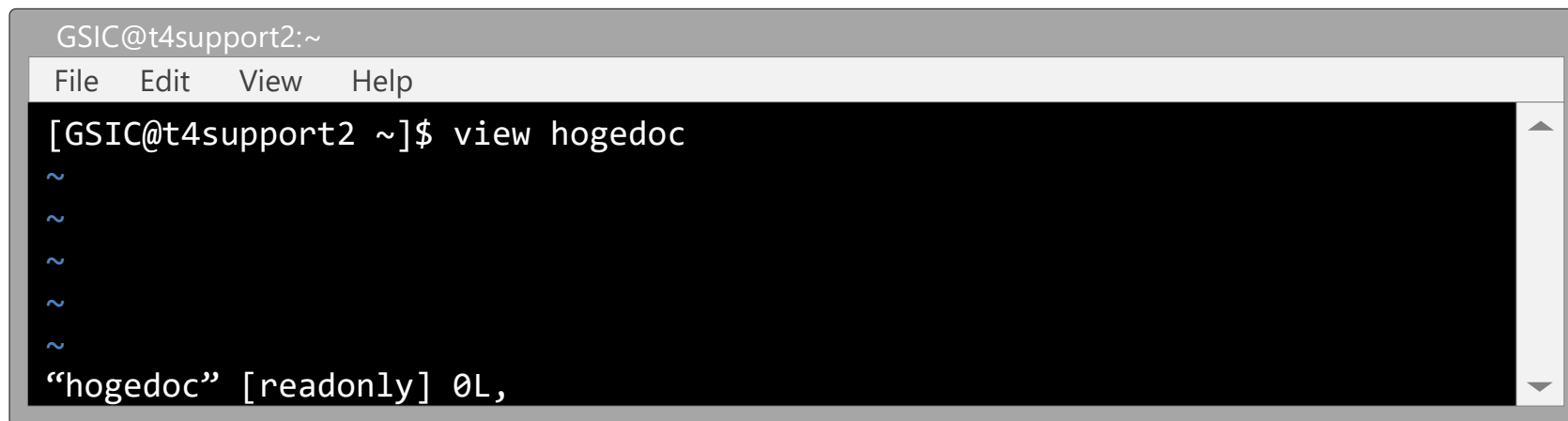
Wipe out all edits and quit (exit vi editor) by :q!

Undo the last change to the file by u

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

view

- An editor command similar to vi editor can be used.
- Open editor in read-only mode.
- Use to check the contents of file.

```
GSIC@t4support2:~
File   Edit   View    Help
[GSIC@t4support2 ~]$ view hogedoc
~
~
~
~
~
"hogedoc" [readonly] 0L,
```

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

- Using cat to view the content of a file
- Using more/less to output the large size files in page

The contents of a file are displayed in page units. page feed is done with the space key.

```
GSIC@t4support2:~
File    Edit    View    Help
[GSIC@t4support2 ~]$ more /usr/share/doc/python-2.7.5/LICENSE
A. HISTORY OF THE SOFTWARE
==========================


Python was created in the early 1990s by Guido van Rossum at Stichting
Mathematisch Centrum (CWI, see http://www.cwi.nl) in the Netherlands
as a successor of a language called ABC.  Guido remains Python's
principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for
National Research Initiatives (CNRI, see http://www.cnri.reston.va.us)
in Reston, Virginia where he released several versions of the
--More--(29%)
```

find

- find is the command for get a file location.

  → It is convenient to use when you know part of file name or whole file name.

  → It is impossible to search within a directory without execution rights.

- There are which and whereis as similar commands

```
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

- To check file type from character string.
- Note that sometimes it is wrongly judged.


```
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

- Searches the character string in file
- 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

Displays the differences between 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);
```

Online manual

- 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)
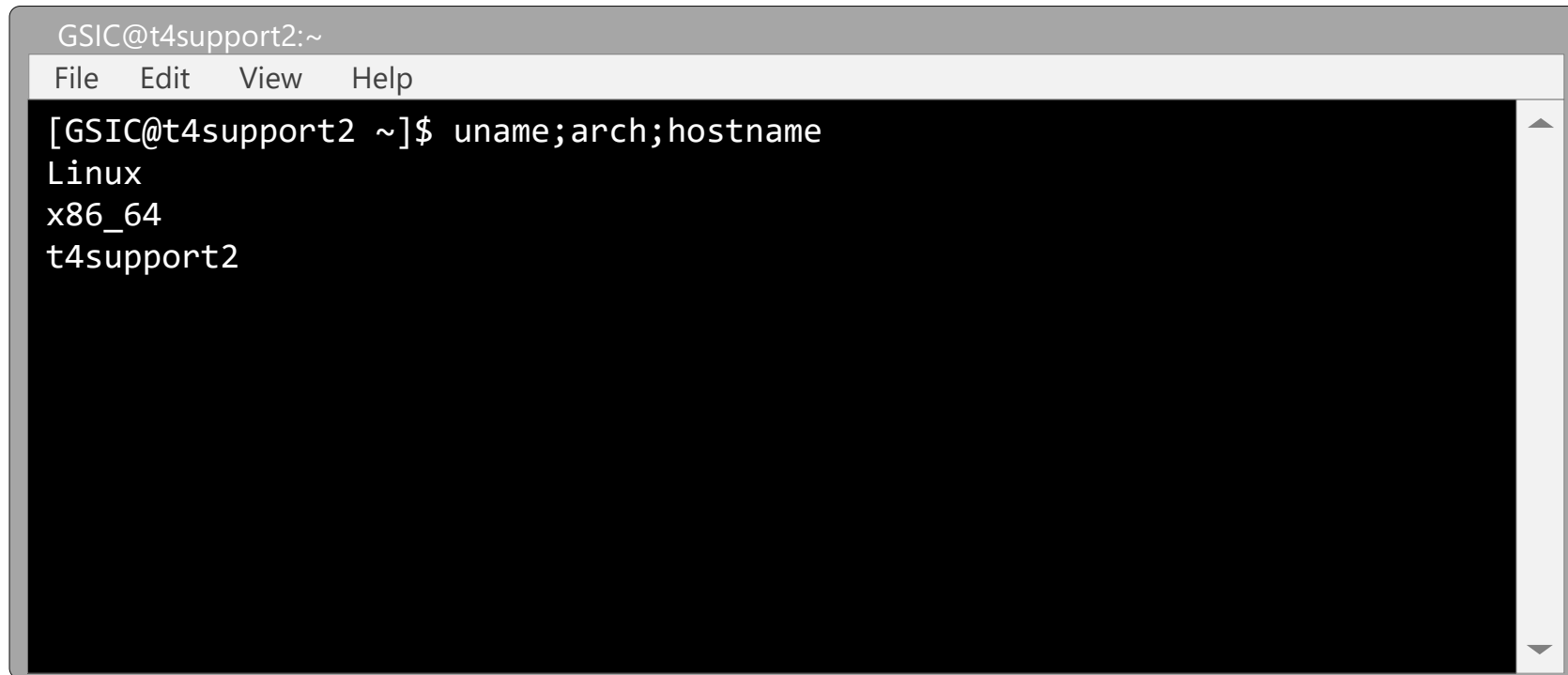
Command concatenation

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.

GSIC@t4support2:~

File   Edit   View   Help

```
[GSIC@t4support2 ~]$ uname;arch;hostname
Linux
x86_64
t4support2
```

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

- \>   Create the stdout obtained with ls to ls.txt.

- \>\>   Append the stdout obtained with ls to ls.txt.

```
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

```
GSIC@t4support2:~
File    Edit    View    Help
[GSIC@t4support2 ~]$ a.out < input.dat
```
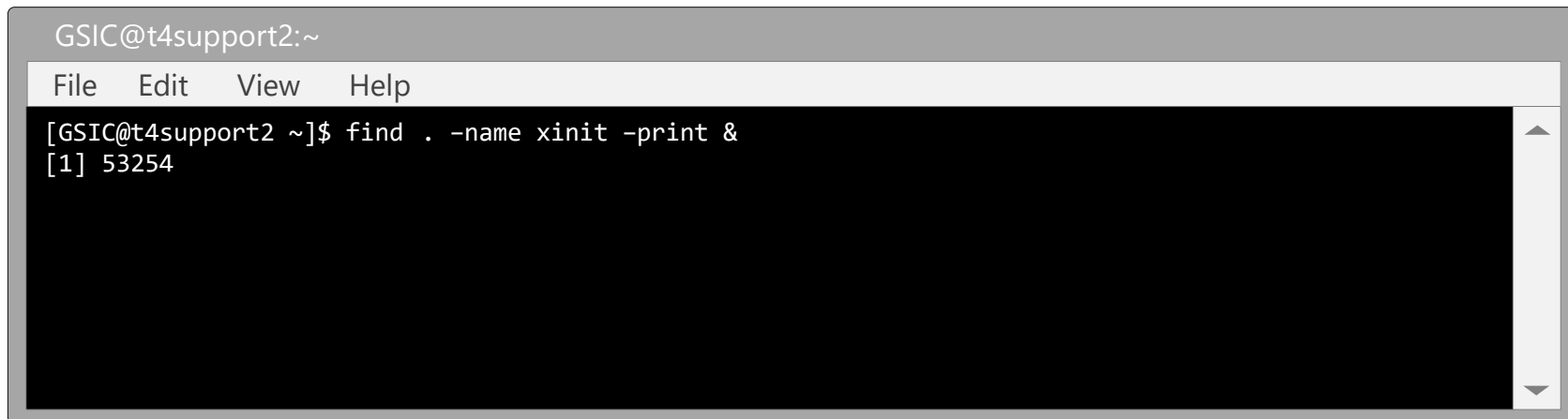
  - Input from standard input

```
GSIC@t4support2:~
File    Edit    View    Help
[GSIC@t4support2 ~]$ a.out << EndOfFile
100
EndOfFile
$
```

Background execution

- Command (program) which takes longer time in execution
- In order 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.

```
GSIC@t4support2:~

File    Edit    View    Help

[GSIC@t4support2 ~]$ find . –name xinit –print &
[1] 53254
```

View running process

- Process is execution unit of processing on OS
  - The following example shows three processes running.

```
GSIC@t4support2:~
File    Edit    View    Help
GSICUSER00@r6i3n2:~> 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
  3207 hpe_use+  20    0  425428  60592    1356 S  2.640 0.092 30:26.49 a.out
```

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.
  - jobs command to check running jobs with jobs command

```
GSIC@t4support2:~

File    Edit    View    Help

[GSIC@t4support2 ~]$ sleep 30 &
[1] 3423
 [GSIC@t4support2 ~]$ jobs
[1]+  Running                 sleep 30 &
[GSIC@t4support2 ~]$
```

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.

qsub/qrsh

Login node → Admin node → Compute node
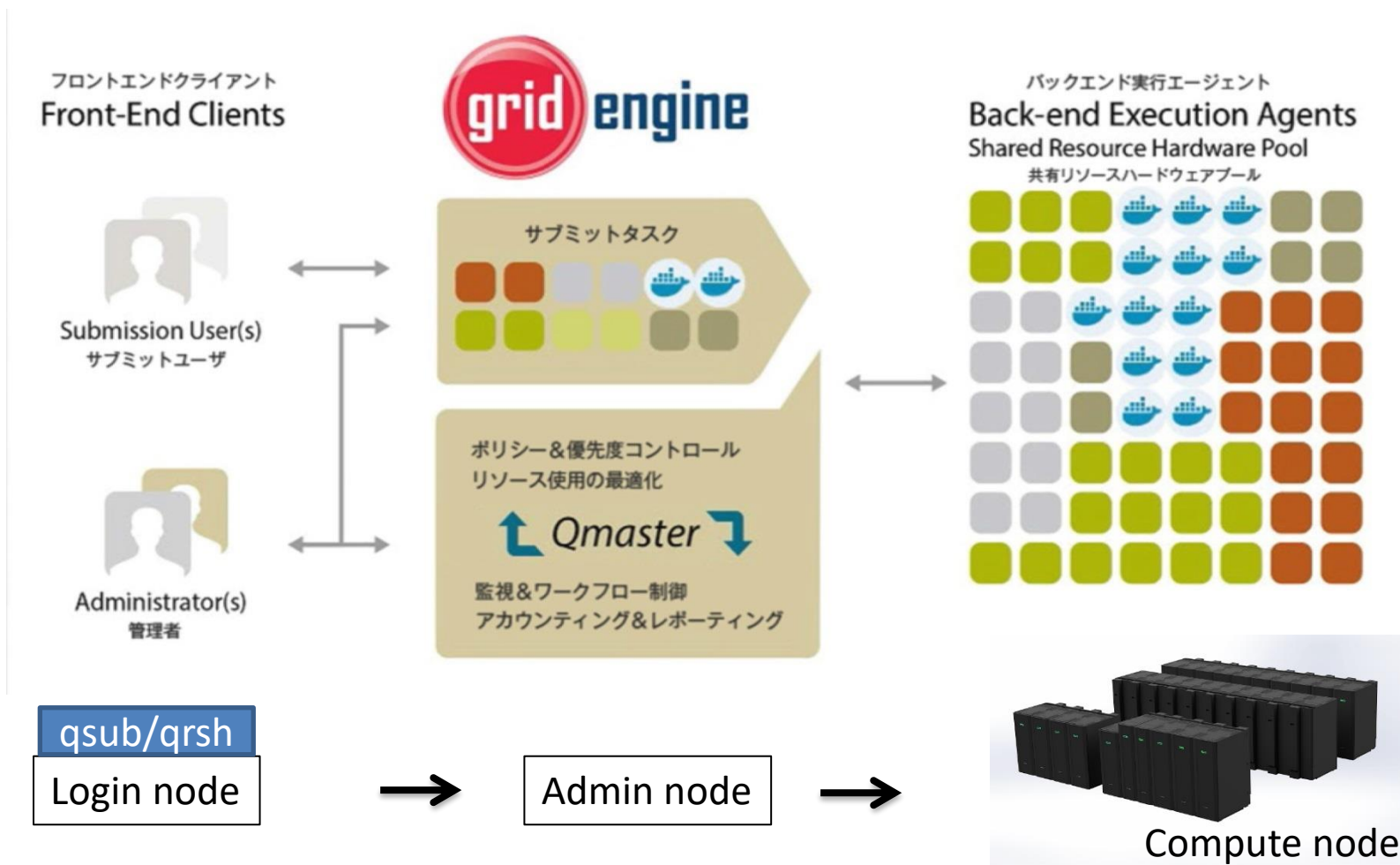
Use Compute node (qsub/qrsh)

Interactive submission

qrsh

Batch submission

qsub

Node reservation
qsub/arsh
with -ar Option

Launch Apps from
Open OnDemand
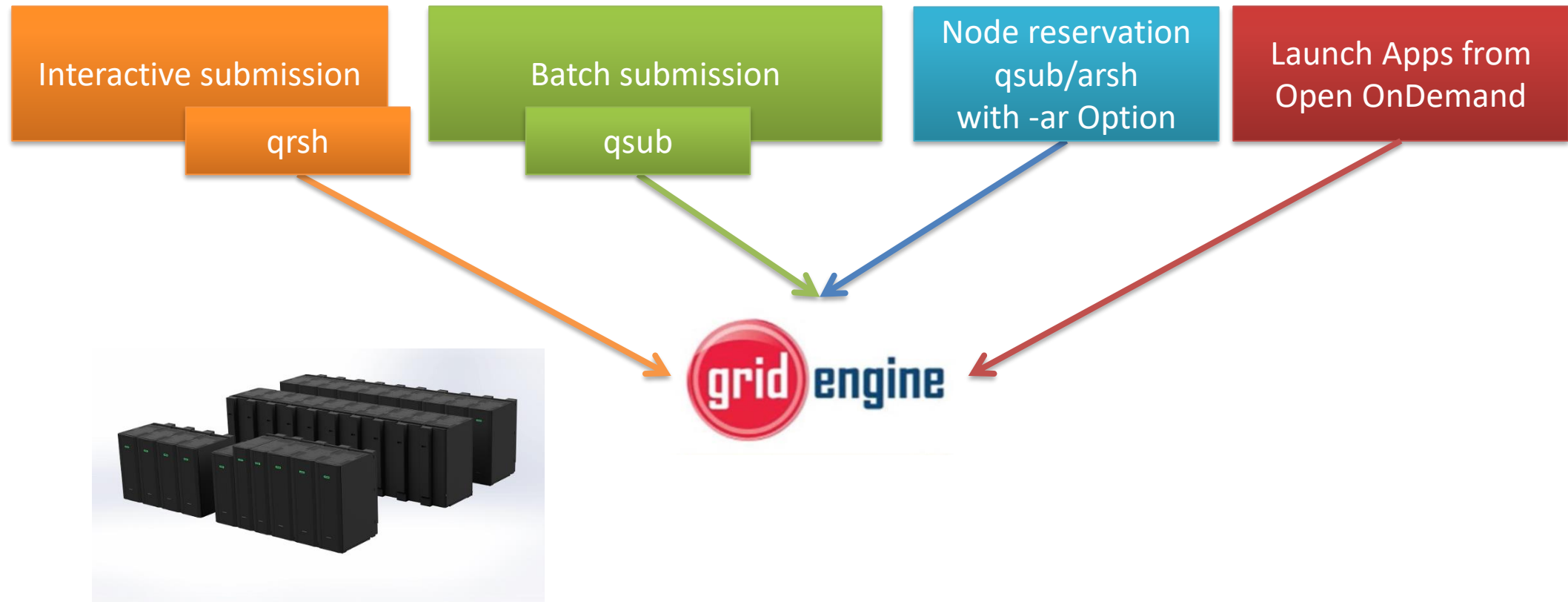
Flow from file creation to submit

- Prepare program you would like to use.
- Consider resouces need to use the job. (num. nodes, run time length)
- Select one of resource types.
  - node_f, node_q, cpu_4, or …
- Create a batch job script for job scheduler.
- Submit the job by qsub.
- Check job status.
- 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 allocates One/multiple node(s) of 240 compute node to a Job. Users can not 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.gsic.titech.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=0:10:0          <- 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.gsic.titech.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 10min 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.gsic.titech.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 10min and 2 nodes.
[GSIC@login1 ~]$ qrsh –l node_f=2 –l h_rt=0:10:0
[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:~>
```

The size of a running program

- Command to measure the memory size of running programs

ps aux, top

*Press q key to stop top command.

Forcely stop a program

- Confirm the process ID of a running program.

  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
```

Process information gathering commands (1)

- Display the users logging in -> *who* command
- Display status of the users logging in -> *w* command

```
GSIC@t4support:~

File    Edit    View    Help

[ux00000@t4support ~]$ who
GSIC     pts/5          2017-09-06 10:52 (:1)
GSIC     pts/6          2017-09-06 11:22 (:1)
TEST     pts/2          2017-09-07 10:05 (192.168.111.63)
GSICUSE pts/7          2017-10-03 12:30 (:2)
GSICUSE pts/3          2017-10-04 12:37 (192.168.111.63)


[ux00000@t4support ~]$ w
 12:39:20 up 51 days,  3:29, 12 users,  load average: 1.00, 1.01, 1.00
USER      TTY       FROM           LOGIN@   IDLE   JCPU    PCPU WHAT
GSIC     pts/5     :1                  0611月10 ?xdm?  22days 22.80s gdm-session-worker
GSIC     pts/6     :1                  0611月10 41days  0.25s 10:00  /usr/libexec/
TEST     pts/2     192.168.111.63   06 9月17 4days  1.26s  1.26s -bash
GSICUSE pts/7     :2                  火12 24:09m  0.05s  0.05s bash
GSICUSE pts/3     192.168.111.62   12:37 0.00s  0.04s  0.00s w
```

- Display the status of a node -> *top* command

```
GSIC@t4support:~

File    Edit    View    Help

[GSIC@t4support ~]$ top
Tasks: 784 total,   1 running, 781 sleeping,   2 stopped,   0 zombie
%Cpu(s):  0.1 us,  0.1 sy,  0.0 ni, 99.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:  26377451+total, 14203016 used, 24957150+free,     3352 buffers
KiB Swap:        0 total,        0 used,        0 free. 10473748 cached Mem
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
225488 root      20   0  371260  20796   4816 S  2.990 0.008 118:58.30 nv-hostengine
  1111 root      20   0  110076  66204  61604 S  0.332 0.025   0:19.26 systemd-journal
```

- Display running processes -> *ps* command

```
GSIC@t4support:~

File    Edit    View    Help

[GSIC@t4support ~]$ ps aux

[GSIC@t4support ~]$ ps aux | grep
ps aux | grep 1111
root        1111  0.0  0.0 110076 66224 ?        Ss   Sep28   0:19 /usr/lib/systemd/systemd-journald
GSIC      352412  0.0  0.0  10240  1640 pts/0    S+   12:29   0:00 grep --color=auto 1111
```

Example (qsub)

- Submit a batch job to TSUBAME (trial run)

  $ qsub sample.sh

  Check job status
  $ qstat

  If ${JOB_NAME}.o${JOB_ID} and ${JOB_NAME}.e${JOB_ID} are created in working directory, the job has already finished.

User environment

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

Shell

- Shell is a computer program that exposes an operating system's 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.

Load 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.

Variables

- Shell variable
  - Variable valid in a current shell.

- Environment variable
  - Variable valid in every shell.

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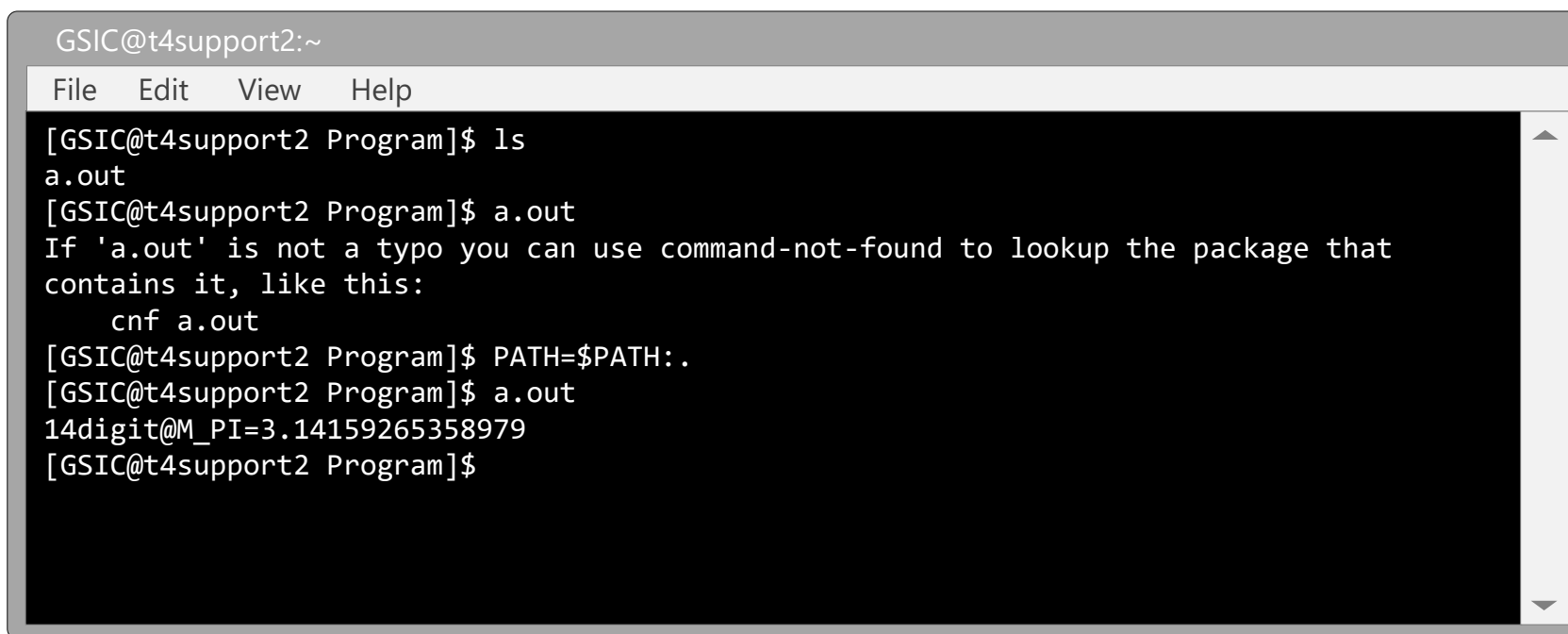
```
534 cd
535 ls
536 history 3
```

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.

GSIC@t4support2:~

File	Edit	View	Help

```
[GSIC@t4support2 Program]$ ls
a.out
[GSIC@t4support2 Program]$ a.out
If 'a.out' is not a typo you can use command-not-found to lookup the package that
contains it, like this:
    cnf a.out
[GSIC@t4support2 Program]$ PATH=$PATH:.
[GSIC@t4support2 Program]$ a.out
14digit@M_PI=3.14159265358979
[GSIC@t4support2 Program]$
```

※ 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 among 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

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
```

SSH connection Examples

- login to TSUBAME4 with login name (GSIC→GSCIUSER00)

```
GSIC@t4support:~

File    Edit    View    Help

[GSIC@t4support ~]$ ssh login.t4.gsic.titech.ac.jp -l ux00000 -i ~/.ssh/id_ecdsa
Last login: Tue Oct  3 09:26:54 2017 from 131.112.3.100
ux00000@login1:~>
ux00000@login1:~> top
ux00000@login1:~> exit
```

- login to TSUBAME4 as the same login name (GSCIUSER→GSCIUSER)

```
GSIC@t4support:~

File    Edit    View    Help

[ux00000@t4support ~]$ ssh login.t4.gsic.titech.ac.jp -i ~/.ssh/id_ecdsa
Last login: Tue Oct  4 09:26:54 2017 from 131.112.3.100
ux00000@login1:~>
ux00000@login1:~> top
ux00000@login1:~> exit
```

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.
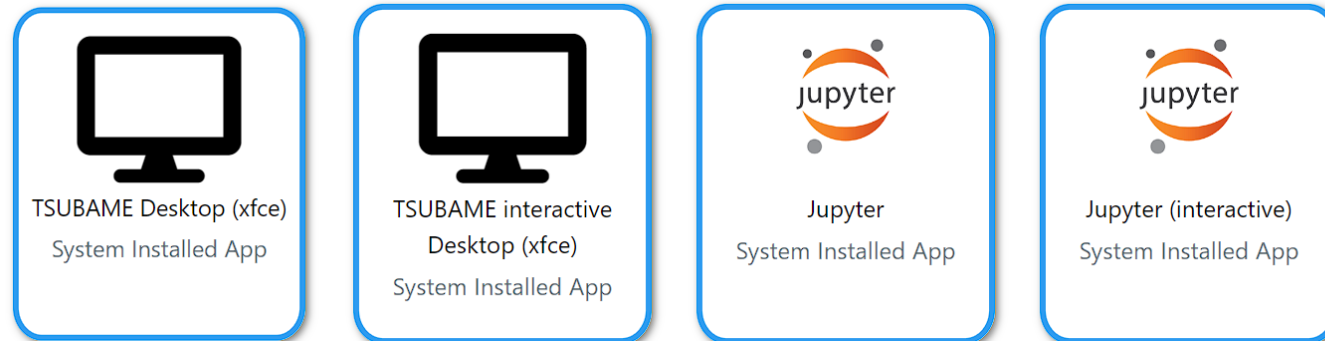   % export DISPLAY=*server_host_name*

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.

Pinned Apps A featured subset of all available apps

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

That's all !
Thank you!

Get more information about T4

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

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