

Szkolenie Użytkowników ICM

3/11/2022

Zespół Oprogramowania i Wsparcia Użytkowników



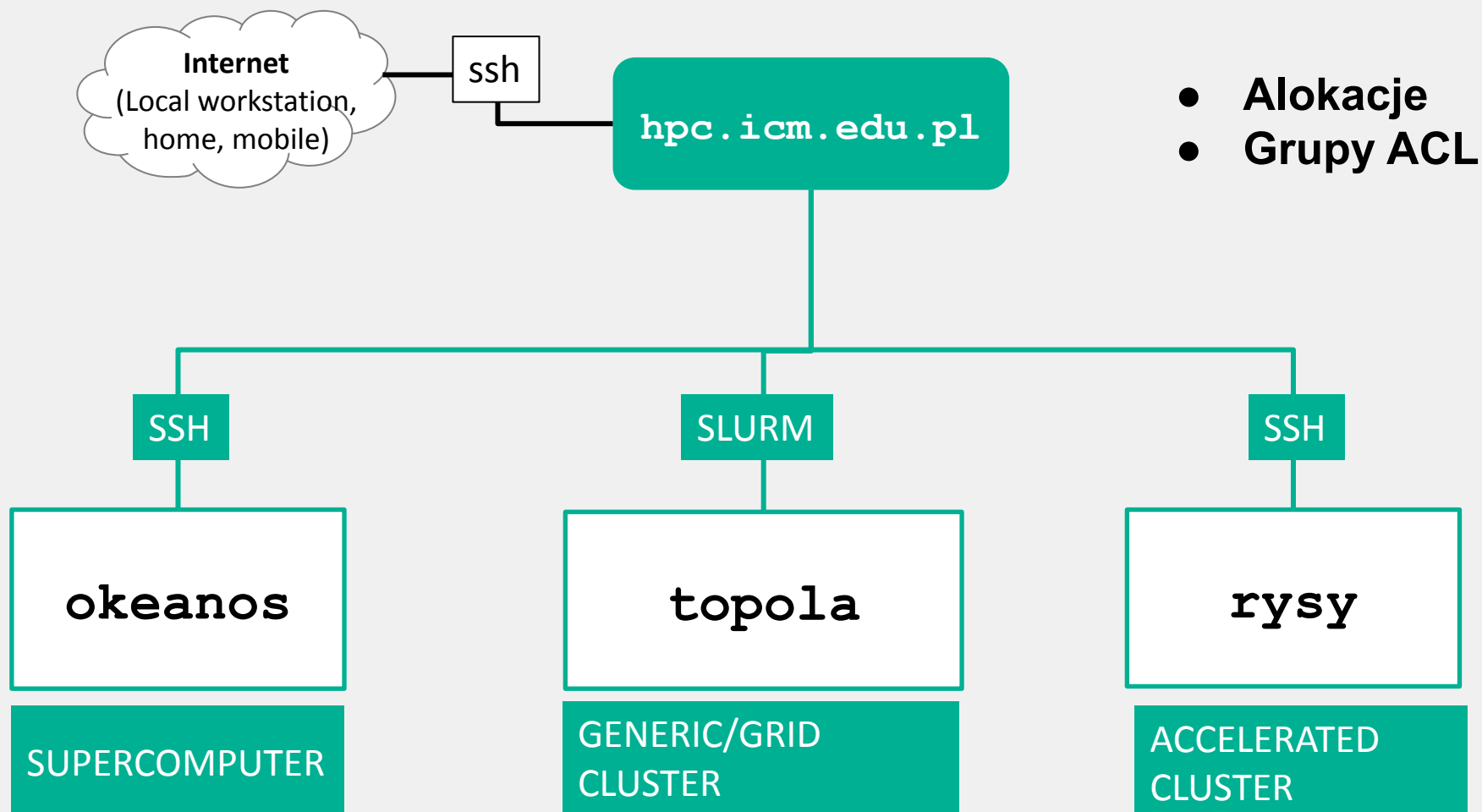
- 1) Wprowadzenie do tematyki szkolenia
- 2) Infrastruktura obliczeniowa ICM UW
- 3) Charakterystyka dostępnych zasobów
 - Topola
 - Okeanos
 - Rysy (GPU/Pbaran)
- 4) Dostęp do systemów
 - Elementy środowiska GNU/Linux
 - Logowanie
 - System plików / kopiowanie danych
 - Edycja / operacje na plikach
 - Aplikacje użytkowe / system modułów
- 5) System kolejkowy SLURM
 - Podstawowe instrukcje
 - Praca interaktywna i wsadowa
 - Przykłady (Topola, Okeanos, Rysy)
- 6) Sesja praktyczna

- ICM świadczy kompleksowe usługi z zakresu obliczeń wielkoskalowych, analityki i wizualizacji danych dla nauki, biznesu i sektora publicznego.
- Oferta ICM obejmuje zarówno rozwiązania standardowe, jak i dostosowane do indywidualnych potrzeb.
- Nasze kluczowe kompetencje to:
 - Obliczenia wielkoskalowe
 - Analityka BigData
 - Modelowanie złożonych procesów
 - wizualizacja
- Świadczymy również usługi polegające na:
 - udostępnianiu infrastruktury obliczeniowej i analitycznej
 - współpracy projektowej: wsparcie techniczne i programistyczne
 - konsultacjach i szkoleniach

Jak zostać użytkownikiem ICM

- Należy wystąpić o **grant obliczeniowy ICM** (<https://granty.icm.edu.pl>)
 - Opis merytoryczny
 - Specyfikacja zasobów (liczba godzin)
 - Procedura recenzji
- Dokumentacja ICM: <http://kdm.icm.edu.pl>
- **Grant obliczeniowy**
 - Temat badawczy, plan obliczeń
 - Kierownik naukowy
 - Wykonawcy grantu
 - Publikacje
 - Raportowanie
- Wsparcie techniczne (pomoc@icm.edu.pl)

Dostęp do systemów obliczeniowych ICM



Okeanos – superkomputer (Cray XC40)



System: Cray XC40

Liczba rdzeni: 26 016

Pamięć: 140 TB

Architektura: Intel Xeon E5-2690 v3 (Haswell) 2.6 GHz

Interconnect: Cray Aries

Rdzeni na węzeł: 24 rdzenie (2x12)

Pamięć na węzeł: 128 GB

System plików: Lustre

System operacyjny: Linux

System kolejkowy: SLURM

Topola – generic cluster (Huawei)



System: Klaster Huawei E9000

Liczba rdzeni: 6244

Pamięć: 18 TB

**Architektura: Intel Xeon E5-2697 v3
(Haswell) 2.1 GHz**

Interconnect: Infiniband FDR

Rdzeni na węzeł: 28 rdzeni (2x14)

Pamięć na węzeł: 64/128 GB

System plików: Lustre

System operacyjny: Linux

System kolejkowy: SLURM

Rysy – klaster GPU / PBaran



Rysy

System: **klaster GPU**

Architektura: Intel Skylake NVIDIA
Volta

Węzły obliczeniowe: 7

Parametry węzła:

- Rdzenie: 36
- RAM: 380 GB
- GPU: 4



PBaran

System: **NEC SX-Aurora Tsubasa**

Architektura:

- VE: SX-Aurora Tsubasa
- VH: Intel Xeon Gold 6126

Węzły obliczeniowe: 1

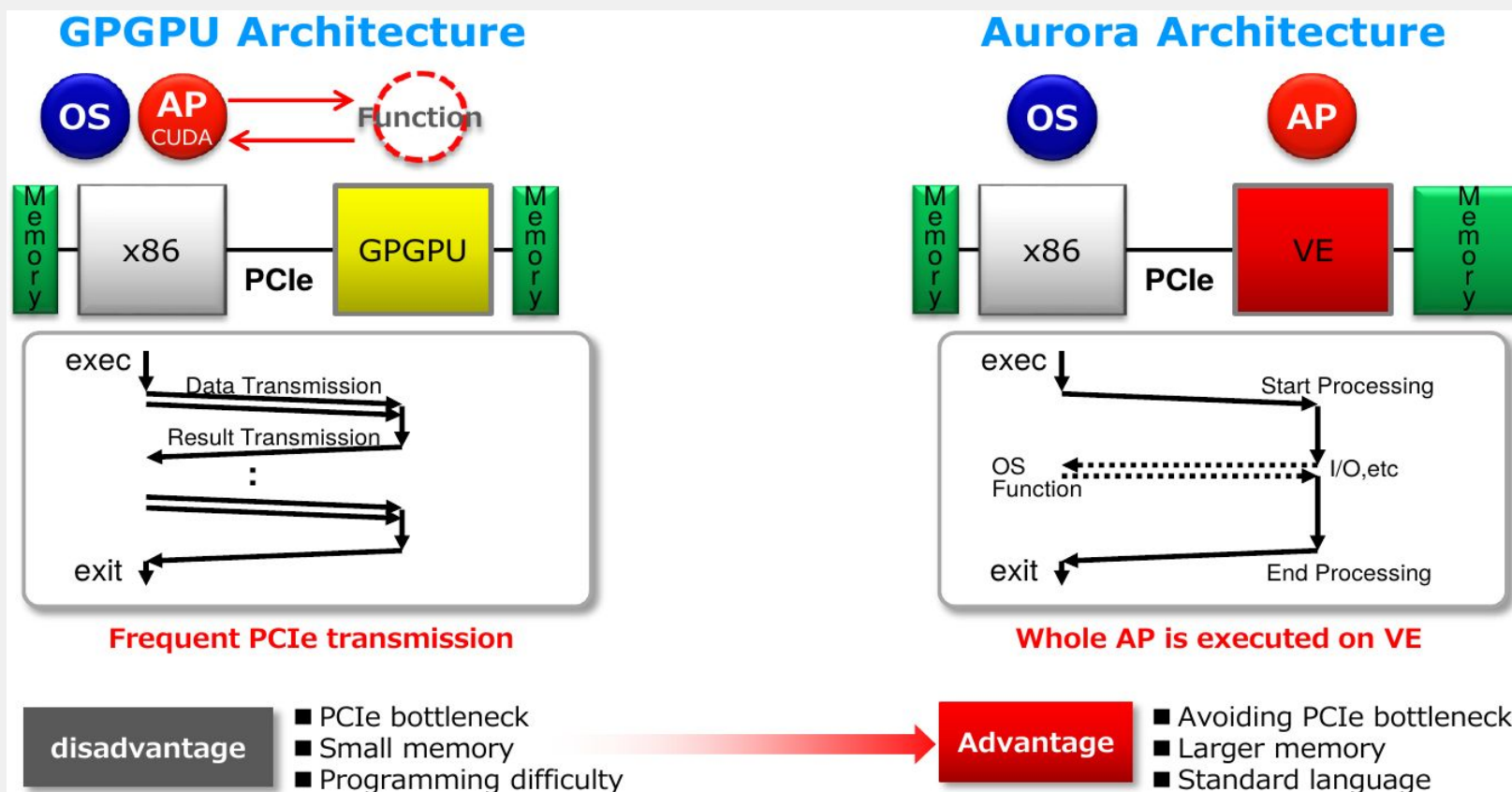
Parametry węzła:

- Rdzenie: 8 x 8 (VE) + 2 x 12 (VH)
- RAM: 8 x 48 GB (VE) + 192 GB (VH)

Oprogramowanie

Chemia obliczeniowa; AI (SOL); biblioteki numeryczne; kompilatory C/C++/Fortran

Rysy – klaster GPU / PBaran



© NEC Corporation

Rysy – klaster GPU / PBaran

Programing Environment



```
$ vi sample.c  
$ ncc sample.c
```

Vector Cross Compiler

automatic vectorization

automatic parallelization

Fortran: F2003, F2008

C/C++: C11/C++14

OpenMP: OpenMP4.5

Library: MPI3.1, libc, BLAS, Lapack, etc

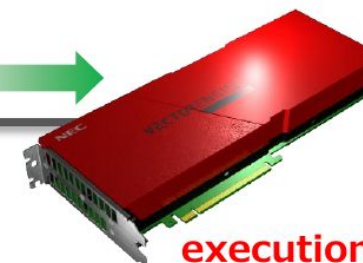
Debugger: gdb, Eclipse parallel tools platform

Tools: PROGINF, Ftrace Viewer

Execution Environment



```
$ a.out
```

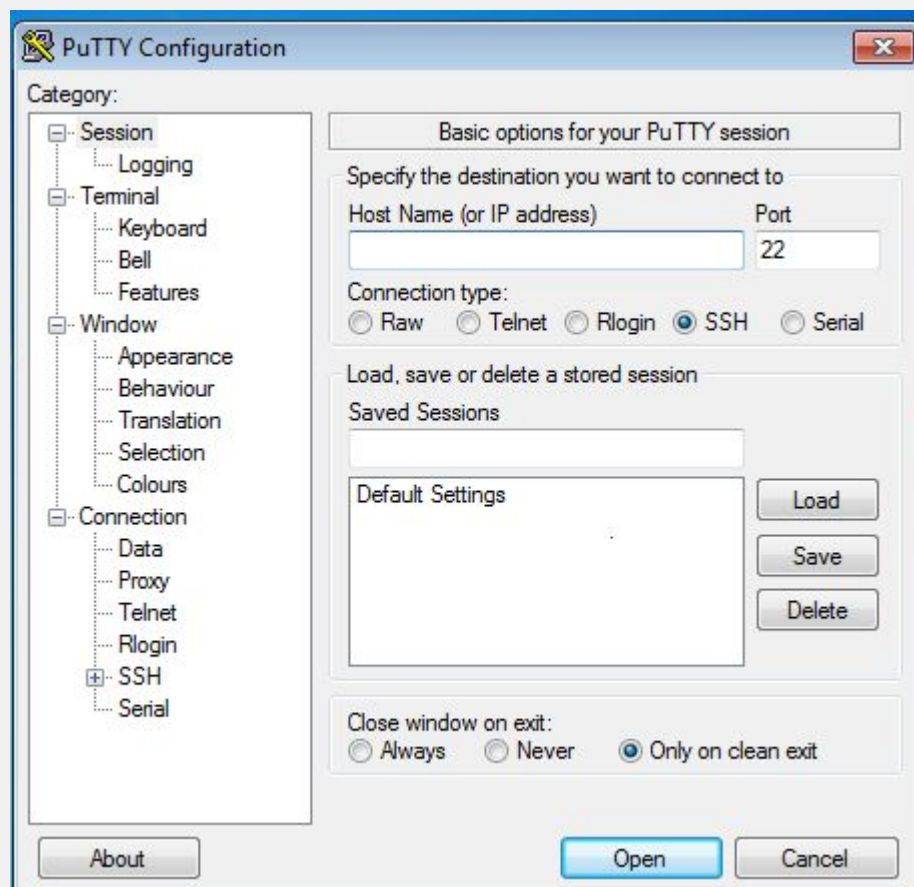


- Logowanie do ICM
 - protokół **SSH**
 - **Putty** (Windows), PowerShell (Windows 10), **ssh** (Linux, Mac)
- Węzeł dostępowy
 - **hpc.icm.edu.pl** (login.icm.edu.pl)
- SSH command:

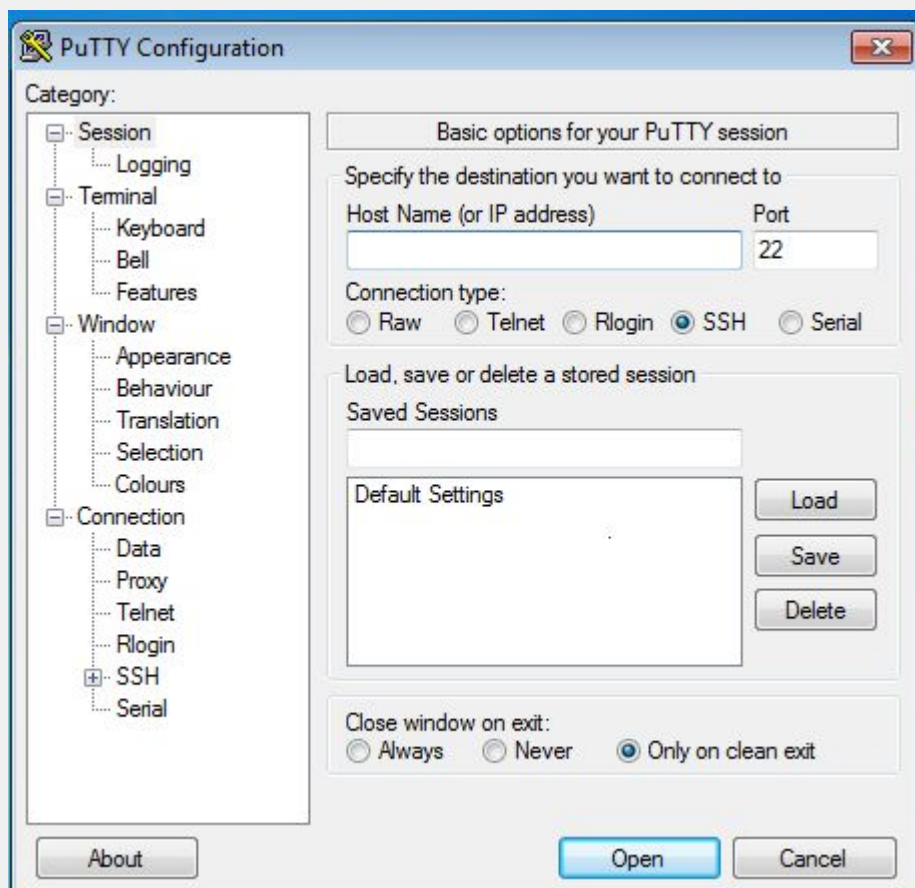
```
ssh user_name@hpc.icm.edu.pl
```

Dostęp do systemów ICM

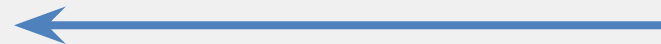
`user_name@hpc.icm.edu.pl`



Dostęp do systemów ICM



CLICK



SYSTEM OPERACYJNY GNU/LINUX

UNIX: wieloużytkowy system operacyjny

Unix - filozofia (Doug McIlroy):

- *write programs that do one thing and do it well,*
- *write programs to work together,*
- *write programs that handle text streams as a universal interface.*

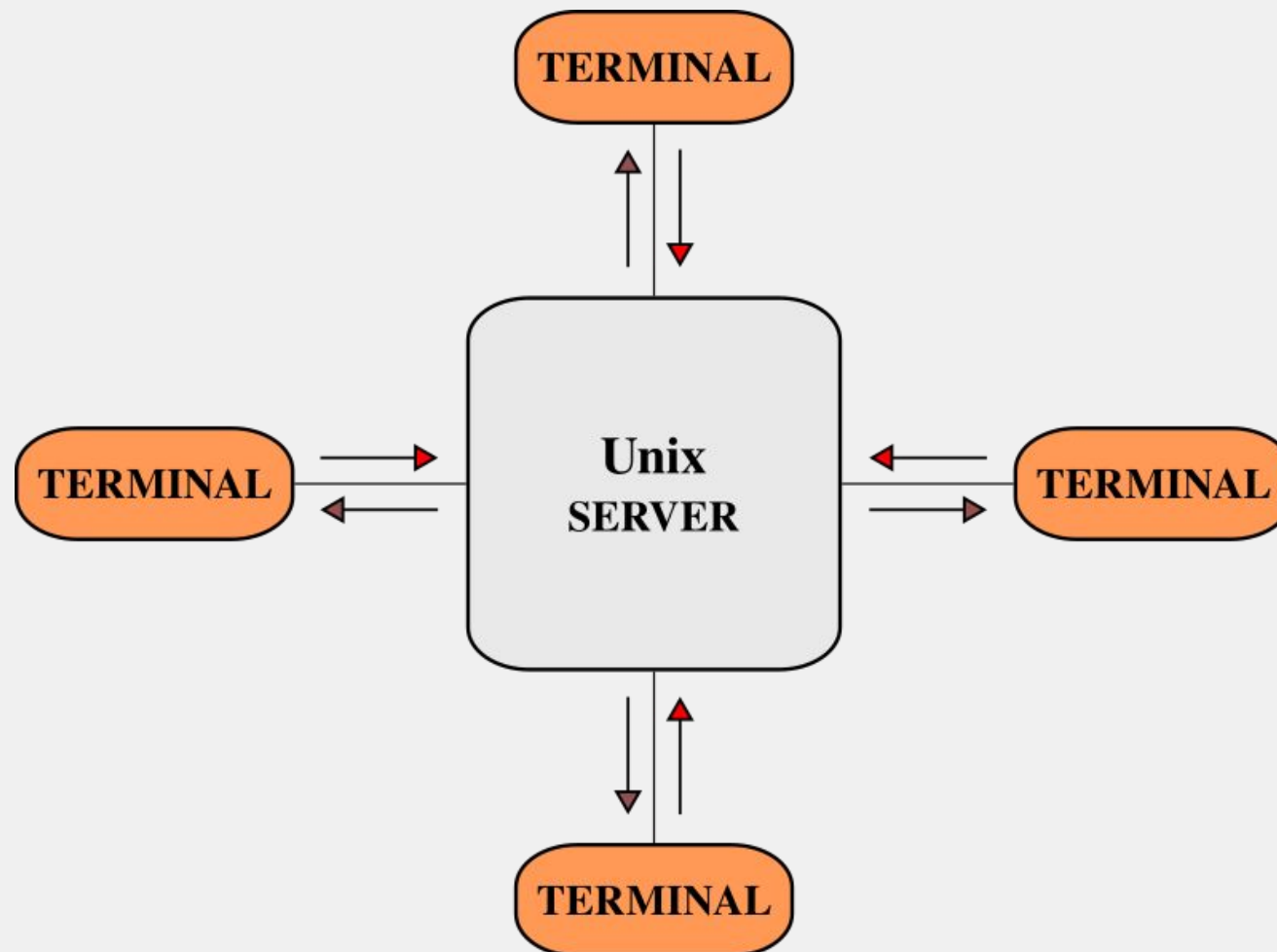
Wybrane cechy systemu Unix:

- wielozadaniowy, wieloużytkowy (timesharing OS),
- monolityczne jądro, hierarchiczny system plików,
- wszystko jest plikiem (nawet urządzenia!),
- koncepcja **pipeline** (**potok**).



Zdjęcie: **Peter Hamer** [CC BY-SA 2.0] via Wikimedia Commons

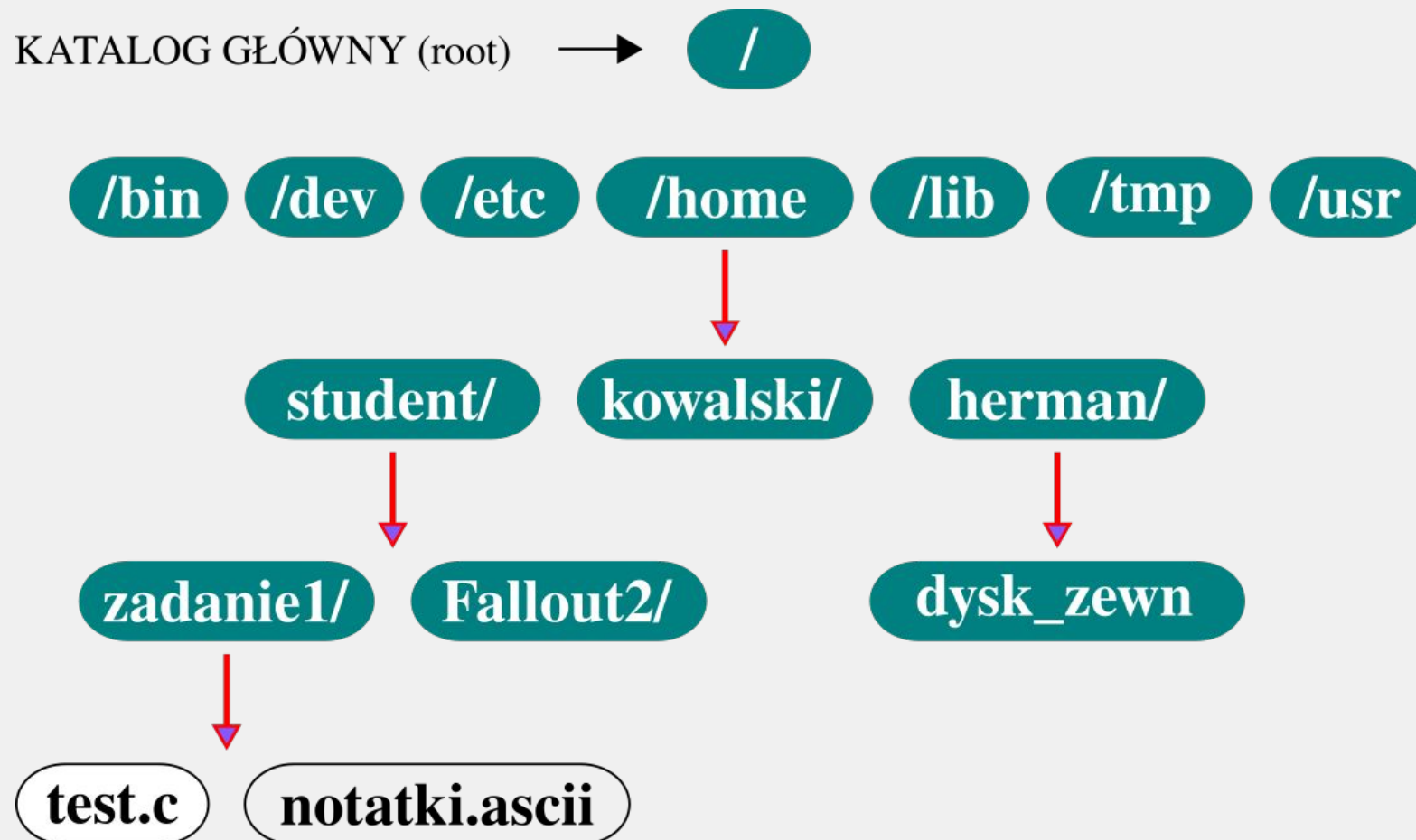
UNIX: wieloużytkowy system operacyjny



- 1983 r. GNU: System operacyjny **JAK UNIX**
- GNU = “Gnu’s Not Unix”
- GNU + Linux – kompletny system operacyjny,
- Dostępny na wiele architektur sprzętowych (amd64, arm64, i386, powerpc, mips, s390x),
- Mnogość dystrybucji systemu: CentOS, Debian, Slackware, RedHat i wiele innych.



O systemie plików



Ścieżka bezwzględna – określa lokalizację pliku względem katalogu głównego (/). Na przykład:

```
/icm/home/user_name
```

```
/home/student/zadanie1/test.c
```

Ścieżka względna – określa lokalizację pliku względem bieżącego katalogu roboczego (. /). Na przykład:

```
zadanie1/test.c
```

```
./gaussian_outs
```

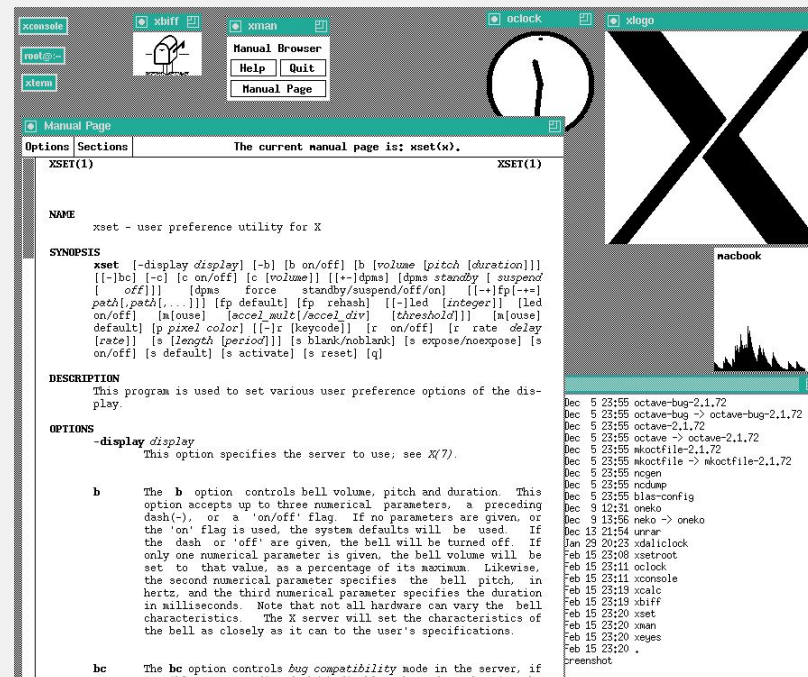
Uwaga na znaki diakrytyczne!

Interfejs użytkownika

```
cron.weekly
csh
csh.cshrc
csh.login
csh.logout
cups
cupshelpers
dbus-1
debconf.conf
debian_version
default
deluser.conf
dhcp
dictionaries-common
discover.conf.d
discover-modprobe.conf
dkms
dpkg
drirc
herman@lepton:/etc$
herman@lepton:/etc$
herman@lepton:/etc$
herman@lepton:/etc$

hosts
hosts.allow
hosts.deny
hp
htdig
i3
i3status.conf
icedove
icedtea-web
iceweasel
idmapd.conf
ifplugd
ImageMagick-6
init
init.d
initramfs-tools
inputrc
insserv
insserv.conf
```

CLI
(Command Line Interface)



GUI
(Graphical User Interface)

Zrzut ekranu z prawej: Liberal Classic [MIT] via Wikimedia Commons

GNU Bash (Bourne Again SHell):

interpreter języka poleceń – powłoka (shell) systemu GNU/Linux.

The GNU Bash Reference Manual, v. 4.3 (<http://www.gnu.org>).

Jest domyślną powłoką systemu GNU/Linux – jego CLI (Command Line Interface), umożliwia pracę interaktywną (wprowadzanie poleceń), a także wsadową (wykonywanie skryptów).

Inne powłoki: sh, csh, tcsh, ksh, zsh.

\$ program [opcje] [argumenty]

`ls` – listuje zawartość bieżącego katalogu roboczego,

`ls path` – listuje zawartość ścieżki *path*

Wybrane opcje:

- `l` permissions, owner, group, size, last-modified date and filename
- `a` lists all files in the given directory
- `d` shows information about a symbolic link or directory
- `t` sort the list of files by modification time.
- `h` print sizes in human readable format.
- `r` revers sorting order

Zmiana katalogu roboczego:

```
cd path_to_directory  
cd test/  
cd /icm/home/kdmszk01/test
```

Przejdźcie do katalogu “wyżej” w hierarchii systemu plików:

```
cd ../
```

Przejdźcie do katalogu domowego:

```
cd $HOME  
or  
cd ~/
```

*W systemach POSIX-owych **wszystko jest plikiem!***

`mkdir directory` – utwórz pusty katalog

`rmdir directory` – usuń pusty katalog

`mkdir -p path_to_directory` – utwórz pełną ścieżkę (hierarchię katalogów)

`rm -r directory` – usuń katalog wraz z zawartością

Uprawnienia plików/katalogów do:

zapisu – write (**w**),

odczytu – read (**r**),

wykonania – execute (**x**).

Dla:

użytkownika – user(**u**),

grupy – group (**g**),

innych – other (**o**),

wszystkich – all (**a**).

Format:

- rwx rwx rwx

`chmod` *target action permission*

Na przykład:

`chmod g+w,o-wx file`

rwx r-x r-x → rwx rwx ---

```
cp źródło cel
```

```
cp path_to_sourcefile path_to_targetfile
```

Wybrane opcje:

- r kopiuje rekursywnie
- p zachowuje datę modyfikacji, uprawnienia etc.

Na przykład:

```
cp out1 out_1
```

```
cp -rp OUTPUTS /tmp/20191113/
```

mv - like cp but remove sourcefile

- * wszystko w bieżącym katalogu
- *.out wszystkie pliki, których nazwa kończy się na **.out**
- a* wszystkie pliki, których nazwa zaczyna się na **a**
- */ * wszystkie pliki znajdujące się w podkatalogach
- ? zastępuje dowolny symbol
- [set] dowolny symbol z podanego zbioru
- [!set] dowolny symbol spoza podanego zbioru

Archiwizacja:

TAR – *Tape ARchive* (archiwum taśmowe)

Kompresja:

ZIP (`zip -r źródło cel`)

GZIP (`gzip źródło.tar`)

BZIP (`bzip2 źródło.tar`)

Przykłady:

```
tar cvzf źródło.tar.gz cel
```

```
tar xvzf archive.tar.gz
```

```
unzip archive.zip
```

`grep string plik` – listuje wiersze zawierające wzorzec "string" w *pliku*
`grep 'Etot' *.out`

Wybrane opcje:

- c – show counts of findens in files
- L – show files without 'str'
- s – skipp errors
- n – show line number wher 'str' was finded
- w – searchinf for whole world
- i – Ignore case distinctions
- r – Read all files under each directory, recursively

`find` – wyszukiwanie plików/katalogów **według nazwy:**

```
find path -name file_name
```

```
find path -type d -name directory_name
```

Znajdź i usuń:

```
find path -type f -name files_name -exec rm -f {} \;
```

Patrz także:

według użytkownika, uprawnień, daty modyfikacji, rozmiaru etc.

Przekierowanie standardowego wyjścia do pliku (**nadpisanie!**):

```
vasp_std > out.log
```

```
vasp_std 1> out.log
```

Przekierowanie standardowego wyjścia do pliku (**DOpisanie!**)

```
vasp_std >> out.log
```

Przekierowanie standardowego wyjścia diagnostycznego do pliku:

```
vasp_std 2> out.log
```

Przekierowanie zawartości pliku input do innego pliku:

```
g18 < input
```

Potok – pipeline, symbol: |

przekierowuje standardowe WYjście (stdout) programu na WEjście innego.

Przykład z użyciem narzędzia grep (**nieefektywne! ale obrazowe**):

```
$ cat liczby.txt | grep 23
```

```
23
```

```
245723
```

```
$ wc -l liczby.txt (← z przełącznikiem -l zlicza wiersze)
```

```
5
```

```
$ cat liczby.txt | grep 23 | wc -l
```

```
2
```

- `cat file` – listuje zawartość pliku
- `more file` – listuje zawartość z podziałem na strony (down-scrolling)
- `less file` – listuje zawartość z podziałem na strony (up/down-scrolling)
- `head -n N file` – wyświetla pierwszych N wierszy pliku
- `tail -n N file` – wyświetla ostatnich N wierszy pliku
- `cut -f N -d "s"` – wyświetla N-tą kolumnę pliku z separatorem "s".
Wymaga argumentu w postaci nazwy pliku lub danych z potoku.

Edycja plików: vim

```
NERD_tree_4 - (~/workspace/whydodiet) - GVIM
<me/jinesh/workspace/whydodiet/ 1 class ExerciseLogsController < ApplicationController
  2 ..
  3 def index
  4   @exercise_logs = current_user.exercise_logs
  5   start = BodyLogEntry::DEFAULT_GRAPH_START_DATE
  6   @chart = LazyHighCharts::HighChart.new('graph') do |f|
  7     f.title(:text => "Exercise Track")
  8     f.x_axis(:type => "datetime")
  9     f.series(
 10       :name => 'Calories Burnt',
 11       :pointInterval => 2.days.to_i * UnitConversions.send("second")["millisecond"],
 12       :pointStart => start.to_i * UnitConversions.send("second")["millisecond"],
 13       :data => current_user.graph_data_on_exercise_log_for(start.to_date..Date.today)
 14     )
 15     yAxes = Array.new
 16     yAxes << {:title => {:text => "In calories"}}
 17     f.options[:y_axis] = yAxes
 18     f.chart()
 19   end
 20 end

N  master > app/controllers/exercise_logs_controller.rb  unix < utf-8 < ruby 1% 1:1
1 %article.diet_list.exhibit 1 class ExerciseLog < ActiveRecord::Base
2 %h1== Listing exercise logs for #{current_user.name} 2   belongs_to :user
3 %table 3
4 %tr 4   validates :name, :presence => true
5 %th Name 5   validates :calories_burnt, :numericality => true, :presence =>
6 %th Time (In minutes) 6   validates :time, :presence => true
7 %th Calories burnt 7   validates :time, :numericality => true, :if => Proc.new {|ex>
8 %th Recorded on 8
9 %th 9   validates :recorded_on, :presence => true
10 10

~/workspace/whydodiet  master > app/views/exercise_logs/index.haml 2% 1:1  master > app/models/exercise_log.rb 2% 1:1
app/models/recipe_database.rb
test/unit/diet_plan_test.rb
app/views/diet_plans/new.haml
test/fixtures/diet_plans.yml
app/helpers/diets_helper.rb
app/views/diets/index.haml
app/views/diets/_form.haml
app/views/diets/edit.haml
app/views/diets/show.haml
app/views/diets/new.haml
mru  files  buf  -  prt < path  ~/workspace/whydodiet
```

NERDTree: Refreshing the root node. This could take a while...

Edycja plików: vim

:w nazwa	– zapis do pliku,
:wq	– zapis i wyjście,
:q!	– wyjście (ignoruj zmiany),
i	– tryb edycji,
[ESC]	– opuść tryb edycji,
u	– cofnij,
<Ctrl-R>	– powtórz,
/wzorzec	– szukaj wzorca (n/N),
!:polecenie	– polec. powłoki,
dd	– usuń bieżący wiersz,

D	– usuń stąd do końca wiersza,
:%s/raz/dwa/g	– znajdź i zastąp (raz→dwa),
:s/raz/dwa/g	– znajdź i zastąp w bieżącym wierszu,
:s/raz/dwa/gc	– znajdź i zastąp (potwierdź),
v	– zaznacz (nawigacja kursorami),
y	– skopiuj zaznaczenie,
d	– wytnij zaznaczenie,
p	– wklej skopiowane.

Edycja plików: GNU Emacs

```
emacs@blueberry
File Edit Options Buffers Tools Emacs-Lisp Help
[Icons] Save Undo [Icons]
(defun hash-table-empty-p (hash-table)
  "Check whether HASH-TABLE is empty (has 0 elements)."
  (zerop (hash-table-count hash-table)))

(defun hash-table-keys (hash-table)
  "Return a list of keys in HASH-TABLE."
  (let ((keys '()))
    (maphash (lambda (k _v) (push k keys)) hash-table)
    keys))

(defun hash-table-values (hash-table)
  "Return a list of values in HASH-TABLE."
  (let ((values '()))
    ...))

-:--- subr-x.el.gz 36% L148 (Emacs-Lisp)
Next: Distrib, Up: dir
(emacs)Top

The Emacs Editor

Emacs is the extensible, customizable, self-documenting real-time
display editor. This manual describes how to edit with Emacs and some
of the ways to customize it; it corresponds to GNU Emacs version
6.0.50.

If you are reading this in Emacs, type 'h' to read a basic
introduction to the Info documentation system.

U:%%- *info* (emacs) Top Top L9 (Info Narrow)
```

www.emacs.org

```
emacs@trisquel-os
File Edit Options Buffers Tools Help
[Icons] Save Undo [Icons]



Welcome to GNU Emacs, one component of the GNU/Linux operating system.

Emacs Tutorial Learn basic keystroke commands
Emacs Guided Tour Overview of Emacs features at gnu.org
View Emacs Manual View the Emacs manual using Info
Absence of Warranty GNU Emacs comes with ABSOLUTELY NO WARRANTY
Copying Conditions Conditions for redistributing and changing Emacs
Ordering Manuals Purchasing printed copies of manuals

To start... Open a File Open Home Directory Customize Startup
To quit a partially entered command, type Control-g.

This is GNU Emacs 24.3.1 (x86_64-unknown-linux-gnu, GTK+ Version 3.4.2)
of 2014-05-07 on trisquel-os
Copyright (C) 2013 Free Software Foundation, Inc.

If an Emacs session crashed recently, type Meta-x recover-session RET
to recover the files you were editing.

U:%%- *GNU Emacs* All L5 (Fundamental)
For information about GNU Emacs and the GNU system, type C-h C-a.
```

MC: Midnight Commander



```
<- ~/szkolenia/CUDA/profile_GPU .[^]> <- /mnt/software .[^]>
'n      Name      Size  Modify time  'n      Name      Size  Modify time
/..      UP--DIR  Jan 23 2019 /..      UP--DIR  Jul 23 14:06
Makefile      348 Nov 3 2016 /.snapshot    2048 Nov 23 2017
NOTATKI       100 Nov 3 2016 /apps         6144 Nov 28 2014
define.h      307 Nov 3 2016 /biomed       2048 Sep 11 2013
energia_kinetyczna.cu 444 Nov 3 2016 /common       6144 Oct 4 2016
energia_kinetyczna.h 157 Nov 3 2016 /dteam        2048 Sep 10 2013
energia_potencjalna.cu 655 Nov 3 2016 /icm          4096 Apr 24 2018
energia_potencjalna.h 168 Nov 3 2016 /modules      2048 Jul 20 2015
kroki.cu       464 Nov 3 2016 /ops          2048 Sep 10 2013
kroki.h        219 Nov 3 2016 /plgrid       4096 May 22 2018
main.cu        803 Nov 3 2016 /private      2048 Jul 23 2015
poczatkowe.cu 1342 Nov 3 2016 /wrf          2048 Jul 21 2017
poczatkowe.h   206 Nov 3 2016
policz_energie.cu 570 Nov 3 2016
policz_energie.h 213 Nov 3 2016
profil         716 Nov 3 2016
przesun_atomy.cu 549 Nov 3 2016
przesun_atomy.h 242 Nov 3 2016
sily.cu        965 Nov 3 2016
sily.h         169 Nov 3 2016
verlet.cu      588 Nov 3 2016
verlet.h       157 Nov 3 2016
wypisz.cu      594 Nov 3 2016
wypisz.h       120 Nov 3 2016
zmien_predkosci.cu 551 Nov 3 2016
zmien_predkosci.h 238 Nov 3 2016

UP--DIR 810G/18446 (0%) UP--DIR 164G/1025G (15%)

Hint: To look at the output of a command in the viewer, use M-!
[memar@hpc software]$
1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9FullDn 10Quit [^]
```

scp: *secure copy*

Copying files **to** ICM (from local computer):

```
scp -C -p plik user_name@hpc.icm.edu.pl:~/
```

Copying files **from** ICM (from local computer):

```
scp -C -p user_name@hpc.icm.edu.pl:~/plik ./
```

-C - with compression

-p - keep last modification date

-r - copying directory

Copying directories:

```
scp -r -C -p directory user_name@hpc.icm.edu.pl:
```

```
scp -r -C -p user_name@hpc.icm.edu.pl:~/directory ./
```

SYSTEM MODUŁÓW

Environment Modules (<https://modules.readthedocs.io/en/v4.1.4>)

The Modules package is a tool that simplify shell initialization and lets users easily modify their environment during the session with modulefiles.

Each modulefile contains the information needed to configure the shell for an application (...) Typically modulefiles instruct the module command to alter or set shell environment variables such as PATH, MANPATH, etc. modulefiles may be shared by many users on a system and users may have their own collection to supplement or replace the shared modulefiles.

Modules can be loaded and unloaded dynamically (...) Modules are useful in managing different versions of applications.

```
#%Module1.0
```

```
module-whatis "This module loads Abinit 8.10.3"  
module load common/compilers/intel
```

```
prereq common/compilers/intel
```

```
set VERSION 8.10.3  
set PREFIX /apps/abinit  
set APP DIR $PREFIX/$VERSION/INTEL/bin
```

```
prepend-path PATH $APP DIR
```


- `module avail`
- `module load ...`
- `module unload ...`
- `module display ...`
- `module -` listuje dostępne opcje

Lokalne moduły użytkownika:

```
module use /path/to/personal/modulefiles
```

Można też:

```
module load use.own
```

– tworzy folder `$HOME/privatemodules` i dodaje go do `MODULEPATH`

SLURM: SYSTEM KOLEJKOWY

SLURM: System kolejkowy

Queue system for supercomputers and cluster
with GNU General Public License

`http://slurm.schedmd.com`

Features:

- allocation of resources (nodes, CPUs, memory) for users jobs
- running parallel jobs on allocated resources queueing jobs
- ensuring equal access to resources



- **node** – single computational unite with CPUs and memory,
- **partition** – set of nodes for user group, special usage or characteristics,
- **job, job step** – single run of numerical package or script,
- **constraint** – characteristics of resources.

SLURM: System kolejkowy

System	Available partitions	Description
okeanos	okeanos (default)	compute nodes
topola	topola	compute nodes
rysy	gpu ve	GPU compute nodes Vector Element node

SLURM: System kolejkowy

salloc – allocate resources for job
sbatch – putting queue script to queue

```
sbatch -n4 my_job.sl
```

scancel – remove job from queue

```
scancel --signal=SIGKILL 1234
```

sinfo – show information about nodes and partitions
squeue – shows queue
srun – run parallel job / script

```
srun -n4 --exclusive moj_program.exe
```

SLURM: System kolejkowy

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST (REASON)
5328766	topola	hfor_LaB	mariana	PD	0:00	10	(Resources)
5328767	topola	hfor_LaB	mariana	PD	0:00	10	(Priority)
5328768	topola	hfor_LaB	mariana	PD	0:00	10	(Priority)
5328769	topola	hfor_LaB	mariana	PD	0:00	10	(Priority)
5329499	topola	snpk8_1	purpurea	R	9:46:09	1	t7-15
5329498	topola	snpk7_5	purpurea	R	9:46:32	1	t7-15
5329497	topola	snpk7_4	purpurea	R	9:46:39	1	t7-15
5329496	topola	snpk7_3	purpurea	R	9:46:45	1	t7-15
5328765	topola	hfor_LaB	mariana	R	30:01	10	t9-[7-8,12],t10-[2-3]

Typowy schemat pracy:

```
srun -p topola --pty bash -l
```

```
module load apps/vasp
```

```
module load common/compilers/intel
```

UWAGA: Kolejność operacji ma znaczenie

SLURM: System kolejkowy

- Praca może odbywać się interaktywnie w terminalu,
- Instrukcje wykonywane bezpośrednio na węźle obliczeniowym,
- Zakończenie sesji oznacza zakończenie zadania.

```
srun -p topola --pty bash -l
```

```
srun --pty --reservation=reserv_name bash -l
```

```
srun --pty -A grant_ID bash -l
```

Tryb wsadowy:

- Skrypt do systemu kolejkowego,
- Wszystkie instrukcje zapisane w skrypcie (tryb nie-interaktywny),
- Parametry zadania kontrolują opcje **sbatch** lub wiersze **#SBATCH** w skrypcie.

```
sbatch -p topola -A icm-szkolenia my_job.sl
```

my_job.sl

```
#!/bin/bash -l  
#SBATCH --time=1  
#SBATCH -N4  
srun hostname |sort
```

- Skrypt jest interpretowany wiersz po wierszu (lista instrukcji do wykonania),
- Skrypt SLURM zawiera wiersze definiujące alokację zasobów – #SBATCH
- Pierwszym wierszem skryptu jest `#!/bin/bash -l`
- Kolejne wiersze to definicje #SBATCH
- Pozostała część skryptu zawiera standardowe instrukcje powłoki
- Komentarze oznaczone znakiem # (**wyjątek: pierwszy wiersz skryptu!**)
- Instrukcje wykonywane na węźle obliczeniowym (w środowisku równoległym) powinny być poprzedzone przez: **srun** lub **mpiexec**

SLURM: System kolejkowy

```
#!/bin/bash -l
#SBATCH --job-name=my_job
#SBATCH --output="my_job.out"
#SBATCH --error="my_job.err"
#SBATCH --account="icm-szkolenia"
#SBATCH --nodes=8
#SBATCH --tasks-per-node=28
#SBATCH --mem=64000
#SBATCH --time=08:00:00
#SBATCH --partition=topola

module load common/mpi/mpich/3.3.1
srun ./my_prog
```

TOPOLA

```
#!/bin/bash -l
#SBATCH --job-name=my_job
#SBATCH --output="my_job.out"
#SBATCH --error="my_job.err"
#SBATCH --account="icm-szkolenia"
#SBATCH --nodes=8
#SBATCH --tasks-per-node=24
#SBATCH --mem=128000
#SBATCH --time=08:00:00
#SBATCH --hint=nomultithread

srun ./my_prog
```

OKEANOS

Nazwa zadania:

`--job-name`

Pamięć przypadająca na jeden węzeł (MB):

`--mem`

`--mem=1024 = 1gb`

Limit walltime:

`--time=hh:mm:ss`

Grant ID:

`-A, --Account`

Przekierowanie standardowego wyjścia lub diagnostyki (stdout, stderr)

`--output, --error`

Liczba węzłów:

`-N, --nodes`

Liczba procesów (MPI):

`-n, --ntasks`

`--ntasks-per-node` (number of processes per node)

Liczba CPU na proces (task) dla MPI+OpenMP:

`--cpus-per-task` (number of CPUs per process)

Partition:

`-p, --partition`

Hyper-threading:

`--hint=[no]multithread`

Zmienne środowiskowe SLURM:

- `SLURM_JOB_ID` – ID zadania
- `SLURM_NODELIST` – lista węzłów
- `SLURM_NTASKS` – liczba procesów w zadaniu,
- `SLURM_TASKS_PER_NODE` – liczba procesów na węzeł.

Topola: Uruchamianie zadania – interaktywnie

1. Logowanie do węzła dostępowego:

```
ssh hpc.icm.edu.pl -l my_login
```

2. Uruchomienie zadania interaktywnego:

```
srun -p topola -A icm-szkolenia --pty bash -l
```

3. (Opcjonalnie) Załadowanie modułu aplikacji lub narzędzia:

```
module load apps/gaussian/g16.A.01
```

4. Uruchomienie obliczeń:

```
g16 < input.dat
```

Topola: Uruchamianie zadania – wsadowo

1. Logowanie do węzła dostępowego:

```
ssh hpc.icm.edu.pl -l my_login
```

2. Utworzenie skryptu SLURM i uruchomienie zadania wsadowo:

```
sbatch job.sl
```

Okeanos i Rysy (GPU/PBaran)

1. Logowanie do węzła dostępowego:

```
ssh hpc.icm.edu.pl -l my_login
```

2. Prze-logowanie do systemu docelowego:

```
ssh okeanos
```

lub

```
ssh rysy
```

3. Uruchomienie zadania – analogicznie jak w przypadku systemu Topola.

KOMPILACJA OPROGRAMOWANIA

Serial code

- Intel: `module load common/compilers/intel`
- GNU: `module load common/compilers/gnu`

Parallel code (MPI):

- Intel + MPICH (recommended), use `mpicc/mpif90`
`module load common/mpi/mpich/3.3.1`
- Intel + Intel MPI (not recommended), use `mpiicc/mpiifort`
`module load intel`
- GNU + OpenMPI, use `mpicc/mpif90`
`module load common/mpi/openmpi/3.0.0`

Serial code

- Cray: default, use cc
- Intel: **module load intel**, use icc
- GNU (gcc): **module load gcc**, use gcc

Parallel code (MPI), use **only cc/ftn**:

- Cray/Cray Clang (NEW): default *note cce module version*
- Intel: **module swap PrgEnv-cray PrgEnv-intel**
- GNU: **module swap PrgEnv-cray PrgEnv-gnu**

CrayMPI: cc/CC/ftn!

Serial/MPI – VH (Vector Host):

Intel

GNU (gcc)

Serial/MPI – VE (Vector Engine):

NEC compilers (C/C++/Fortran)

Na węźle obliczeniowym (pbaran):

```
source /opt/nec/ve/mpi/2.0.0/bin/necmpivars.sh
```

Rysy/PBaran: Kompilacja oprogramowania



Serial C/C++/Fortran code:

```
ncc a.c
```

```
nc++ a.cpp
```

```
nfort a.f90
```

MPI C/C++/Fortran code:

```
mpincc a.c
```

```
mpinc++ a.cpp
```

```
mpinfort a.f90
```


Running serial code **explicitly**:

```
ve_exec ./a.out
```

Running serial code **implicitly**:

```
./a.out
```

Running MPI code:

```
mpirun ./a.out
```

- Remember to **source** the `necmpivars.sh` (or `necmpivars.csh`)!
- `ve_exec` is located in `/opt/nec/ve/bin`
- Mind that some VE tools installed in the above path are named just like standard GNU/Linux tools

SESJA PRAKTYCZNA

<http://tiny.cc/3gk0vz>

Dziękujemy za uwagę!

<https://kdm.icm.edu.pl>

pomoc@icm.edu.pl

