Кластер ИВМ РАН

Login, компиляция, запуск на счет

И. Н. Коньшин

Содержание

Кластер ИВМ РАН

В ИВМ РАН установлен вычислительный кластер на базе 36 вычислительных узлов с 1440 процессорными ядрами под управлением операционной системы SUSE Linux Enterprise High Performance Computing 15 SP4. Пиковая производительность кластера Rpeak=92,16 Тфлопс. Суммарный объём распределенной оперативной памяти – 9,5 ТБ. Вычислительные узлы кластера объединены высокопроизводительной сетью InfiniBand EDR.

> В январе 2023 года на кластере была обновлена операционная система, см. подробнее.

Кластер ИВМ РАН

http://cluster2.inm.ras.ru

Система очередей **Slurm** кластера ИВМ РАН:

- normal (36 узлов 2x20)
- short (2 узла 2x12)

Узел **normal:**

- Compute Node Arbyte Alkazar R2Q50 G5
- 40 ядер (два 20-ядерных Intel Xeon Gold 6230@2.10 ГГц)
- Оперативная память: 256 Гб

Узел **short:**

- Compute Node Arbyte Alkazar R1Tw50SM G5
- 24 ядра (два 12-ядерных Intel Xeon Silver 4214@2.20 ГГц)
- Оперативная память: 128 Гб

Операционная система: SUSE Linux Enterprise HPC 15 SP4 (x86_64)
Компиляторы C: gcc v. 7.5.0 (2017) или Intel icc v. 2021.8.0

Login на кластер ИВМ РАН

- Если вы работаете из Windows, то нажимаете <Пуск>, потом <cmd>, и в образовавшемся окошке набираете все следующие команды.
- Заходим на промежуточный компьютер внутри ИВМ РАН:
  ```
  ssh visitor@dodo.inm.ras.ru
  Information for guests at dodo server:
  https://dodo.inm.ras.ru/guests
  (visitor@dodo.inm.ras.ru) Password: ********
  ```
- С промежуточного компьютера заходим на сам кластер:
  ```
  ssh student@cluster2.inm.ras.ru
  ```
  или просто:
  ```
  c2
  student@cluster2.inm.ras.ru's password: *******
  ```
- Далее смотрим инструкцию по работе на кластере ИВМ РАН:
  ```
  cat yy-xxx/README.txt
  ```
  или открываем в браузере ссылку с таким же файлом:
  http://cluster2.inm.ras.ru/~student/yy-xxx/README.txt

Краткая инструкция по работе на кластере ИВМ РАН

=== Cluster ===

- `cp -a yy-xxx 2Y-FIO`
  ## copy test directory to your own with `year` and `FIO` or `nick`
- `cd !$`
  ## change directory to your own one
- `mpiicc mpi_hello.c`
  ## compile test file with MPI
- `vi qs_tst`
  ## edit the job file to modify job name
- `sbatch qs_tst`
  ## query substitute for the job file
- `slurmtop`
  ## the top list of executed jobs
- `cat res`
  ## view the result file
- `rmoe`
  ## script to remove slurm-*.out job files
- `mc`
  ## midnight commander: cd, copy, edit, etc.

Копирование

- `cp -a yy-xxx YY-FIO`
  ## copy test directory to your own with `year` and `FIO` or `nickname`

  Вместо `y` здесь вы пишите последнюю цифру текущего года, а вместо `FIO` — свои `инициалы` или *nickname* если вы из ИВМ, МФТИ или Сеченовского университета, и свой `номер в группе` если вы из Университета Сириус

  `cp` (copy) — команда копирования

  `-a` (all) — здесь (и во всех других командах тоже!) "`-`" (минус) означает, что далее идет "опция" (вариант) самой команды, а опция "`a`" означает, что мы копируем файлы, сохраняя "все" атрибуты копируемых файлов

  Таким образом, эта команда образует вашу собственную директорию (папку) `YY-FIO` и скопирует туда все файлы из исходной директории, где лежат много нужных нам для работы файлов (образцы программ, образцы скриптов, инструкции по запуску и т.д.)

Ваша директория

- `cd !$`
  `## change directory to your own one`

  Теперь переходим в вашу директорию.

  `cd` (change directory) — переход в указанную директорию

  `!$` — означает последний набранный вами аргумент в последней вашей команде (в нашем случае — это как раз имя вашей директории `YY-FIO`)

  P.S. Чтобы посмотреть список файлов в своей директории наберите команду `ls` (короткий список) или `ll` (длинный список)

Компилирование

- `mpiicc mpi_hello.c`
  ## compile test file with MPI

  Компилируем программу с помощью MPI.

  `mpiicc` – название MPI (Message Passing Interface) компилятора языка C (C compiler)

  `mpi_hello.c` – название компилируемой программы, одной из тех, которые уже находились в исходной директории, а теперь находятся в вашей; если понадобится, то свой файл вы можете изменять

  P.S.
  `a.out` – такое имя будет присвоено исполняемому файлу вашей программы по умолчанию

Редактирование скрипта

- `vi qs_tst`
  ## edit the job file to modify job name

  Теперь нужно посмотреть файл и немного его отредактировать.

  `vi` — название редактора `vi` (или `vim`)
  `qs_tst` — имя файла, в котором находится скрипт для запуска команды на счет, в котором вместо `FIO` нужно написать именно ваши `FIO`, `инициалы`, *nickname* или свой `номер в группе`, соответственно

  P.S. (редактирование в `vi`):
  - чтобы двигаться по файлу используйте стрелочки
  - подведите курсор к надписи `FIO`
  - нажмите `i` или кнопку `<Insert>` для перехода в режим вставки
  - напишите свои `FIO`
  - нажмите `<Esc>` чтобы выйти из режима вставки
  - наберите `:wq` чтобы выйти из редактора; здесь `:` означает переход в специальный командный режим, `w` означает запись (write) измененных данных, `q` означает выход (quit) из редактора

Скрипт qs_tst

`#!/bin/bash`

– какой именно использовать интерпретатор команд

`#SBATCH --job-name=YY-FIO  ## <-- change year and FIO`

– имя вашего задания; т.к. мы работаем от имени одного пользователя `student`, то `FIO` нам нужно, чтобы различать задания

`#SBATCH --partition short`

– название раздела; в этом разделе ограничение в 1 час на исполнение задания, но зато он почти всегда свободен

`#SBATCH --ntasks=4`

– нам требуется `4` вычислительных ядра на выполнение нашего задания

`#SBATCH --time=10:00`

– ограничиваем времени выполнения задания, если оно вдруг зависнет

`#SBATCH --chdir=.`

– перед выполнением задания перейти в текущую директорию

`mpirun ./a.out > res`

`mpirun` – эта системная программа будет запускать MPI приложение

`./a.out` – абсолютное имя исполняемого файла вашей программы

`> res` – перенаправление (`>`) вывода вашей программы в файл `res`

Постановка задания в очередь

- `sbatch qs_tst`
  ## query substitute for the job file

  Постановка задания в очередь:

  `sbatch` – команда постановки задания в очередь (substitute in batch)

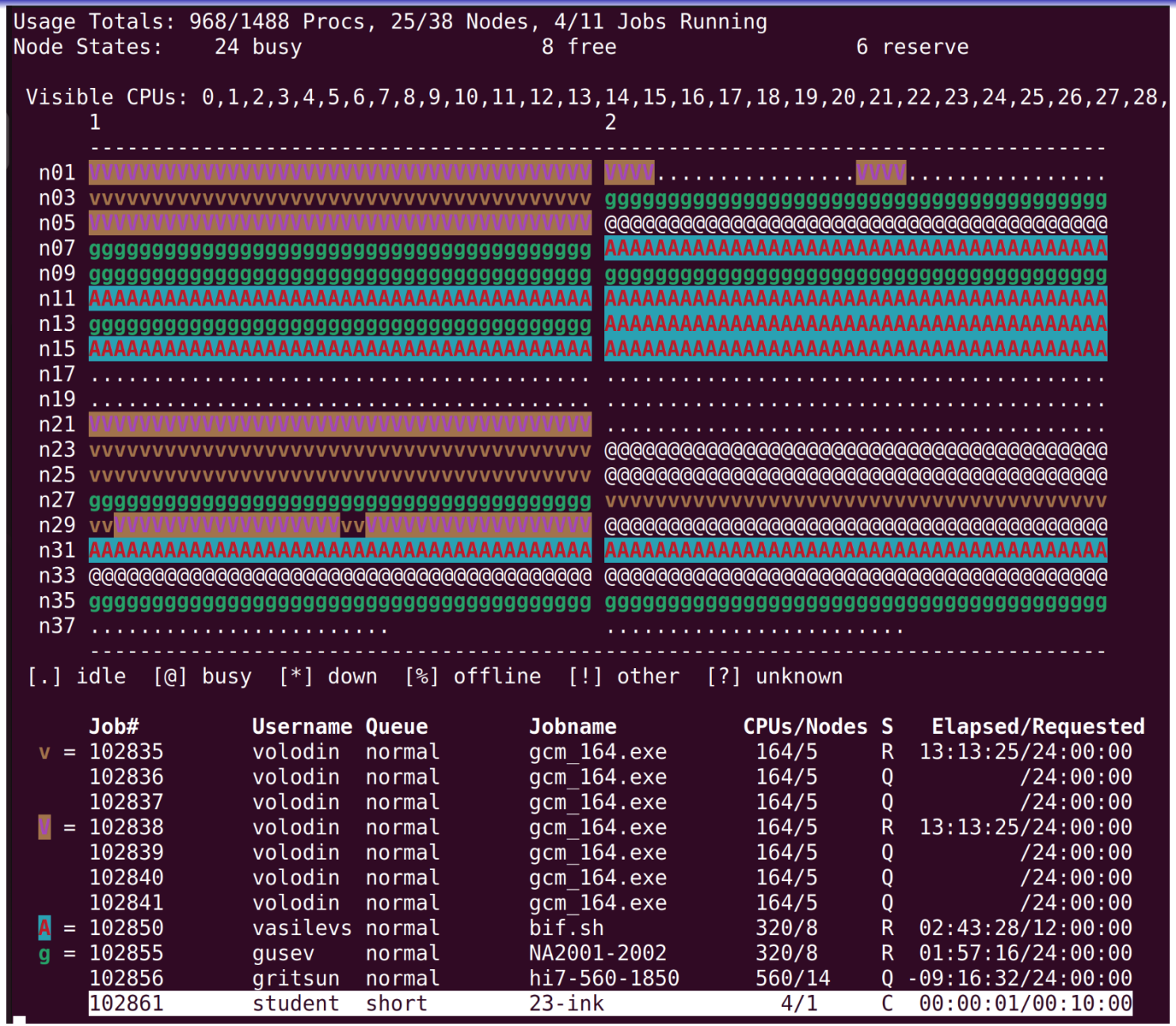
  `qs_tst` – имя файла с командами задания

  Выдача после постановки задания в очередь с номером задания:
  `Submitted batch job 102860`

- `slurmtop`
  ## the top list of executed jobs

  Посмотреть на исполнение заданий и занятость узлов кластера

slurmtop


Usage Totals: 968/1488 Procs, 25/38 Nodes, 4/11 Jobs Running
Node States: 24 busy 8 free 6 reserve
Visible CPUs: 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,
1 2
n01
n03
n05
n07
n09
n11
n13
n15
n17
n19
n21
n23
n25
n27
n29
n31
n33
n35
n37
[.] idle [@] busy [*] down [%] offline [!] other [?] unknown
Job# Username Queue Jobname CPUs/Nodes S Elapsed/Requested
v = 102835 volodin normal gcm_164.exe 164/5 R 13:13:25/24:00:00
102836 volodin normal gcm_164.exe 164/5 Q /24:00:00
102837 volodin normal gcm_164.exe 164/5 Q /24:00:00
V = 102838 volodin normal gcm_164.exe 164/5 R 13:13:25/24:00:00
102839 volodin normal gcm_164.exe 164/5 Q /24:00:00
102840 volodin normal gcm_164.exe 164/5 Q /24:00:00
102841 volodin normal gcm_164.exe 164/5 Q /24:00:00
A = 102850 vasilevs normal bif.sh 320/8 R 02:43:28/12:00:00
g = 102855 gusev normal NA2001-2002 320/8 R 01:57:16/24:00:00
102856 gritsun normal hi7-560-1850 560/14 Q -09:16:32/24:00:00
102861 student short 23-ink 4/1 C 00:00:01/00:10:00

slurmtop

- `nXX` – номер узла, “наши” узлы `n37` и `n38`
- `A` – буква показывает, на каких ядрах выполняется задание

- `Job#` – номер задания
- `Username` – имя пользователя (у нас у всех `student`)
- `Queue` – название очереди (“наша” очередь `short`)
- `Jobname` – имя задания (“наши” имена `2*-***`)
- `CPUs/Nodes` – количество заказанных ядер/узлов
- `S` – состояние (state) или статус (status) выполнения задания: `Q` ждет очереди (query), `R` исполняется (running), `C` завершено (completed, выделяется белым фоном)
- `Elapsed/Requested` – использованное время (если отрицательно, то сколько времени осталось до запуска) и запрошенное время

Результат выполнения

- `cat res`
  ## view the result file

  Посмотреть результат выполнения задания:

  `cat` – вывести на экран содержимое файла (concatenate)

  `res` – имя файла с результатом выполнения

  Пример выдачи:

```
Hello, world, from [2/4]
Hello, world, from [3/4]
Hello, world, from [1/4]
Hello, world, from [0/4]
```

Удалить старые файлы с выдачей

- rmoe

  ## script to remove slurm-*.out job file

  Это синоним более длинной команды, удаляющей все начинающиеся на `slurm-`. файлы с расширением `.out`:
  `alias rmoe='rm slurm-[0-9]*.out'`
  Выполните команду `ll` и найдите свой slurm-*.out файл.
  Этот файл у вас будет иметь длину `0`, т.к. мы писали `> res`.

  Пример выдачи команды `ll`:

```
student@desert: /23-ink> ll
total 100
-rw-r--r-- 1 student users   370 Dec 10  2020 mpi_hello.c
-rw-r--r-- 1 student users  1098 Dec 10  2020 mix_daxpy.c
-rw-r--r-- 1 student users   702 Dec 10  2020 omp_daxpy.c
-rw-r--r-- 1 student users  8823 Feb 19  2021 Notes_mpi_utf.txt
-rw-r--r-- 1 student users   973 Apr 23  2021 mpi_daxpy.c
-rw-r--r-- 1 student users  1085 Feb 10  2022 mpi_norm.c
-rw-r--r-- 1 student users  1177 Dec 16  2022 mix_norm.c
-rw-r--r-- 1 student users   772 Dec 16  2022 omp_norm.c
-rw-r--r-- 1 student users   310 Sep 30 23:23 qs_mpi
-rw-r--r-- 1 student users   536 Sep 30 23:23 qs_mix
-rw-r--r-- 1 student users   324 Sep 30 23:23 qs_omp
-rw-r--r-- 1 student users   214 Sep 30 23:24 qs_tst
-rw-r--r-- 1 student users  5809 Sep 30 23:29 Notes_omp_utf.txt
-rw-r--r-- 1 student users  1419 Oct  1 12:48 README.txt
-rwxr-xr-x 1 student users 30536 Oct  1 14:40 a.out
-rw-r--r-- 1 student users   100 Oct  1 14:40 res
-rw-r--r-- 1 student users     0 Oct  1 13:40 slurm-102861.out
```

mc (Midnight Commander)

- mc

  ## midnight commander: cd, copy, edit, etc.

  Это двухоконный графический интерфейс для работы с файлами.

1 `Help` – подсказка

2 `Menu` – меню

3 `View` – просмотр файла (без возможности редактирования)

4 `Edit` – редактирование файла (в простом редакторе)

5 `Copy` – копирование файла (команда `cp`)

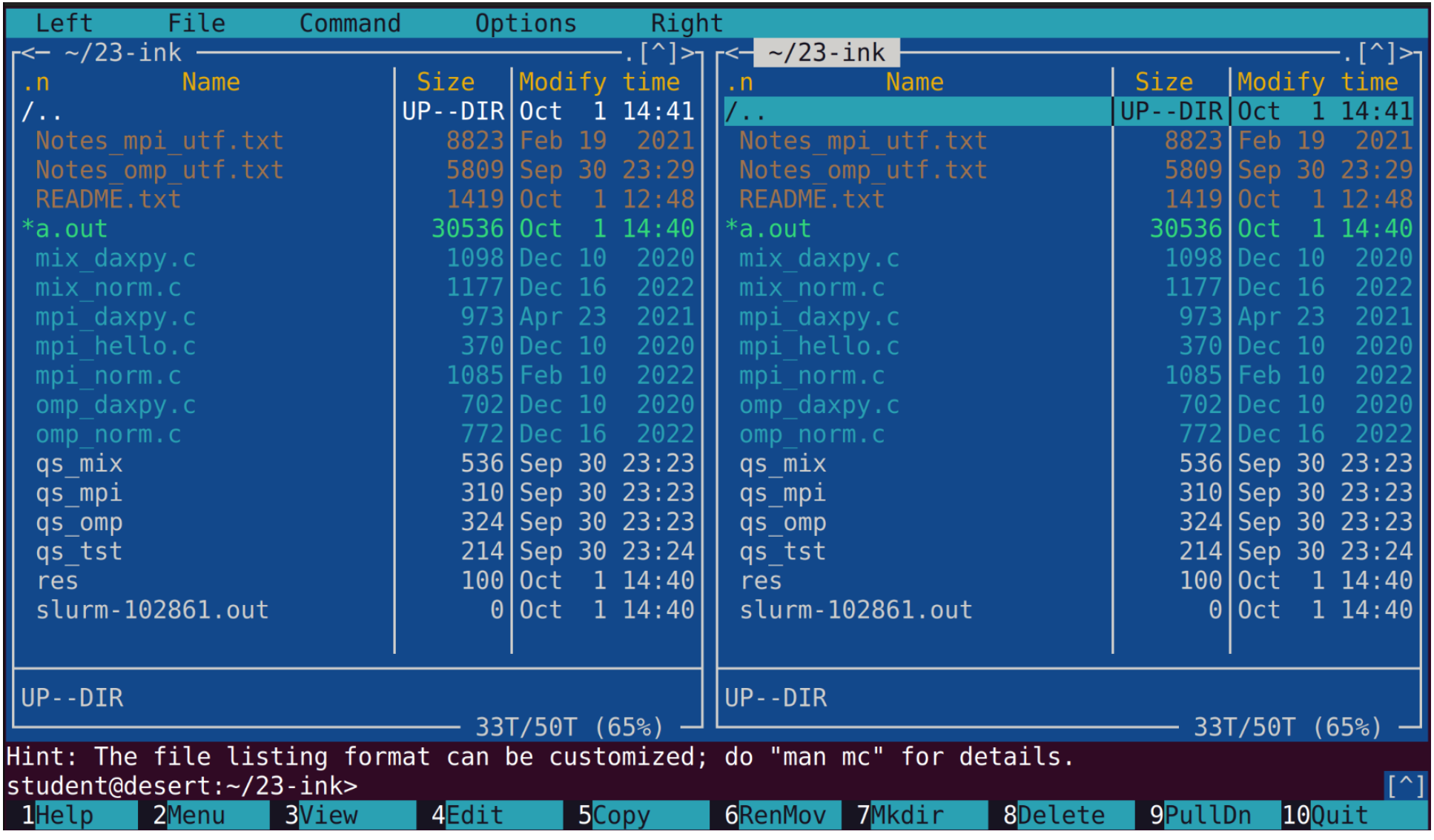
6 `RenMov` – переименование или перемещение файла (команда `mv`)

7 `Mkdir` – создание новой директории (команда `mkdir`)

8 `Delete` – удаление файла (команда `rm`)

9 `PullDn` – специальные продвинутые меню пользователя

10 `Quit` – выход из `mc` с подтверждением


Left File Command Options Right
<- ~/23-ink .[^]>
.n Name Size Modify time
/.. UP--DIR Oct 1 14:41
Notes_mpi_utf.txt 8823 Feb 19 2021
Notes_omp_utf.txt 5809 Sep 30 23:29
README.txt 1419 Oct 1 12:48
*a.out 30536 Oct 1 14:40
mix_daxpy.c 1098 Dec 10 2020
mix_norm.c 1177 Dec 16 2022
mpi_daxpy.c 973 Apr 23 2021
mpi_hello.c 370 Dec 10 2020
mpi_norm.c 1085 Feb 10 2022
omp_daxpy.c 702 Dec 10 2020
omp_norm.c 772 Dec 16 2022
qs_mix 536 Sep 30 23:23
qs_mpi 310 Sep 30 23:23
qs_omp 324 Sep 30 23:23
qs_tst 214 Sep 30 23:24
res 100 Oct 1 14:40
slurm-102861.out 0 Oct 1 14:40
UP--DIR
33T/50T (65%)
<- ~/23-ink .[^]>
.n Name Size Modify time
/.. UP--DIR Oct 1 14:41
Notes_mpi_utf.txt 8823 Feb 19 2021
Notes_omp_utf.txt 5809 Sep 30 23:29
README.txt 1419 Oct 1 12:48
*a.out 30536 Oct 1 14:40
mix_daxpy.c 1098 Dec 10 2020
mix_norm.c 1177 Dec 16 2022
mpi_daxpy.c 973 Apr 23 2021
mpi_hello.c 370 Dec 10 2020
mpi_norm.c 1085 Feb 10 2022
omp_daxpy.c 702 Dec 10 2020
omp_norm.c 772 Dec 16 2022
qs_mix 536 Sep 30 23:23
qs_mpi 310 Sep 30 23:23
qs_omp 324 Sep 30 23:23
qs_tst 214 Sep 30 23:24
res 100 Oct 1 14:40
slurm-102861.out 0 Oct 1 14:40
UP--DIR
33T/50T (65%)
Hint: The file listing format can be customized; do "man mc" for details.
student@desert:~/23-ink> [^]
1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn 10Quit

Перенос данных на кластер и обратно

Можно набрать текст программы в редакторе (например, `vim`) прямо на кластере, но есть и другие варианты...
Набрать его на локальном компьютере (например, в блокноте в Windows), и перенести данные **на кластер** одним из следующих способов:

1. Вставить текст мышкой (Copy/Past) в редакторе (например, `vim`) прямо на кластере)
2. Вставить текст мышкой (Copy/Past) командами Linux без использования редакторов
3. Скопировать файл на свою площадку на внешнем сайте, а потом забрать его на кластере с помощью `wget`, `git`, `svn` или аналогичным образом
4. Скопировать файл с помощью команды `scp` сначала на `visitor@dodo`, а потом забрать его на кластере с помощью той же команды `scp`

Чтобы забрать файл **с кластера**:
скопируйте его в директорию `~/public_html` и заберите его в браузере по адресу https://cluster2.inm.ras.ru/~student/

На кластер [2/4]

`echo` или `cat`

- наберите текст программы в своем любимом редакторе на Windows (блокнот?)
- скопируйте весь текст в буфер обмена (Copy)
- если в тексте программы НЕТ символа `'` (кавычка), то:
  - в Линуксе наберите команду: `echo ' <Enter>`
  - потом мышкой вставьте текст: (Past)
  - и завершите команду echo: `' > myfile <Enter>`
- если в тексте программы ЕСТЬ символ `'` (кавычка), то:
  - в Линуксе наберите команду: `cat > 4.cpp <Enter>`
  - потом мышкой вставьте текст: (Past)
  - и завершите команду cat: `<Enter><Ctrl-D><Enter>`
- у вас на Линуксе образуется файл с нужным текстом :-)
- если потребуется сделать изменения – можно внести изменения у себя на Windows и повторить процедуру

Связанные с этим темы:

- Команды Linux
- Редактор текстов Vim
- Распараллеливание в OpenMP
- Распараллеливание в MPI
- Выполнение практической работы