

## **1. Информация для пользователей раздела KNL**

1.1. Под управление СУППЗ переведен раздел KNL суперкомпьютера [MBC-10П МП2 KNL](#). Особенностью работы раздела KNL является то, что после прохождения очереди СУППЗ задание будет перенаправлено в систему slurm. Пользователи, ранее работавшие со slurm, смогут использовать ее возможности при работе с разделом KNL через СУППЗ. На разделе KNL работают все команды СУППЗ, за исключением `getnodes` и `mrunf`.

## **2. Настройка окружения (выполняется при каждом соединении!)**

2.1. Соединиться с сервером доступа `mvs10q.jscs.ru`

2.2. Для работы с разделом KNL рекомендуется загрузить модуль `kn1` командой

```
module load mpi/intel/kn1
```

2.3. Для работы с СУППЗ необходимо загрузить модуль `launcher/suppz` командой

```
module load launcher/suppz
```

2.4. Убедиться, что модули загружены, можно с помощью команды `module list`

## **3. Запуск программ на разделе KNL**

Запуск программ на разделе KNL осуществляется с помощью стандартных команд СУППЗ. При этом во всех командах СУППЗ нужно указывать ключ `-s kn1`. Приведем примеры типичных команд.

### **3.1. Просмотр очереди**

```
mqinfo -s kn1
```

### **3.2. Запуск задания**

```
mpirun -s kn1 -np 56 mpi_ora
```

При запуске задания на разделе KNL к имени задания автоматически добавится имя логической системы `kn1`. Для нашего примера имя задания будет `kn1.cpi_ora.1`.

### **3.3. Завершение задания**

```
mkill -s kn1 kn1.cpi_ora.1
```

#### **4. Использование логической системы СУППЗ KNL в качестве постоянной**

Если предполагается постоянная работа на разделе KNL, указание ключа `-s knl` в каждой команде СУППЗ может оказаться неудобным. Чтобы логическая система KNL считалась системой по умолчанию, в пользовательском конфигурационном файле СУППЗ `~/.crunmvs` в секцию `[General]` необходимо внести параметр `system` со значением `kn1`. Например, файл можно исправить в редакторе:

```
[General]
host = queue
system = kn1
```

Либо воспользоваться командой

```
confwrite ~/.crunmvs General system kn1
```

Убедиться, что команда сработала правильно, можно, просмотрев конфигурационный файл СУППЗ командой

```
cat ~/.crunmvs
```

После этого указывать ключ `-s kn1` в каждой команде СУППЗ будет не нужно, однако появится необходимость указывать ключ `-s broadwell` при обращении к разделу Broadwell, `-s haswell` – при обращении к разделу Haswell.

#### **5. Использование некоторых возможностей slurm**

Раздел KNL настроен таким образом, что задание проходит через очередь СУППЗ, а затем на выделенных модулях запускается на выполнение через `slurm`. Такой механизм запуска заданий позволяет использовать на разделе KNL командные файлы сценариев работы со `slurm` (`slurm`-скрипты). Команда СУППЗ `mpirun` использует для запуска заданий команду `srun`, а `mbatch` использует для запуска заданий команду `sbatch`. Запуск заданий на разделе KNL возможен только через очередь СУППЗ.

При подготовке slurm-скриптов следует учитывать, что значения параметров `--ntasks`, `--nodes`, `--ntasks-per-node`, `--odelist`, `--output`, `--error` задаются командами СУППЗ (`mpirun` или `mbatch`) и не могут быть изменены в slurm-скрипте. Результат выполнения задания и диагностика ошибок размещаются в каталоге ввода-вывода задания СУППЗ.

Ниже приведены примеры slurm-скрипта и примеры запуска задач с помощью команд `mpirun` и `mbatch` на разделе KNL.

Пример slurm-скрипта:

```
#!/bin/bash
#SBATCH --mem=2G
export OMP_NUM_THREADS=2
./omp_example
```

Пример запуска MPI-приложения `spi_ora` на 32-х процессорах на 10 минут:

```
mpirun -s knl -np 32 -maxtime 10 spi_ora
```

Пример запуска двух процессов MPI-приложения `spi_ora` по одному на вычислительный модуль на 10 минут:

```
mpirun -s knl -np 2 -ppn 1 -maxtime 10 spi_ora
```

Пример запуска 32-х экземпляров slurm-скрипта `run.sh` на 10 минут:

```
mbatch -s knl -np 32 -maxtime 10 run.sh
```

Пример запуска двух экземпляров slurm-скрипта `run.sh` по одному на вычислительный модуль на 10 минут:

```
mbatch -s knl -np 2 -ppn 1 -maxtime 10 run.sh
```