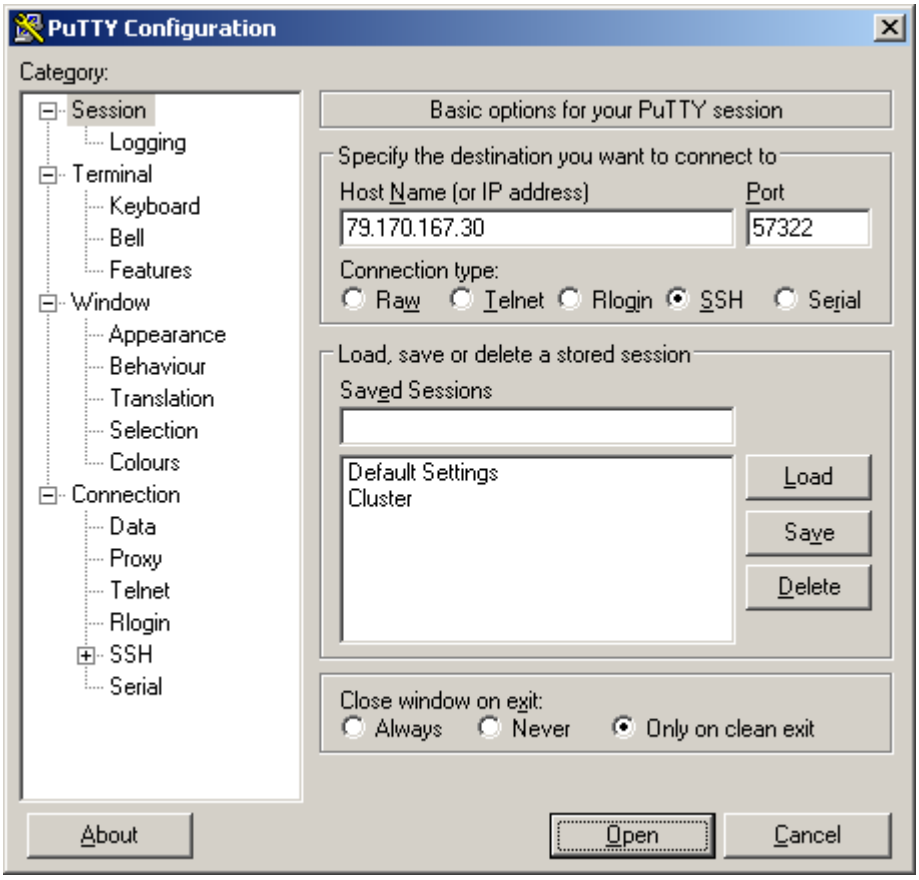
Робошкола – 2017

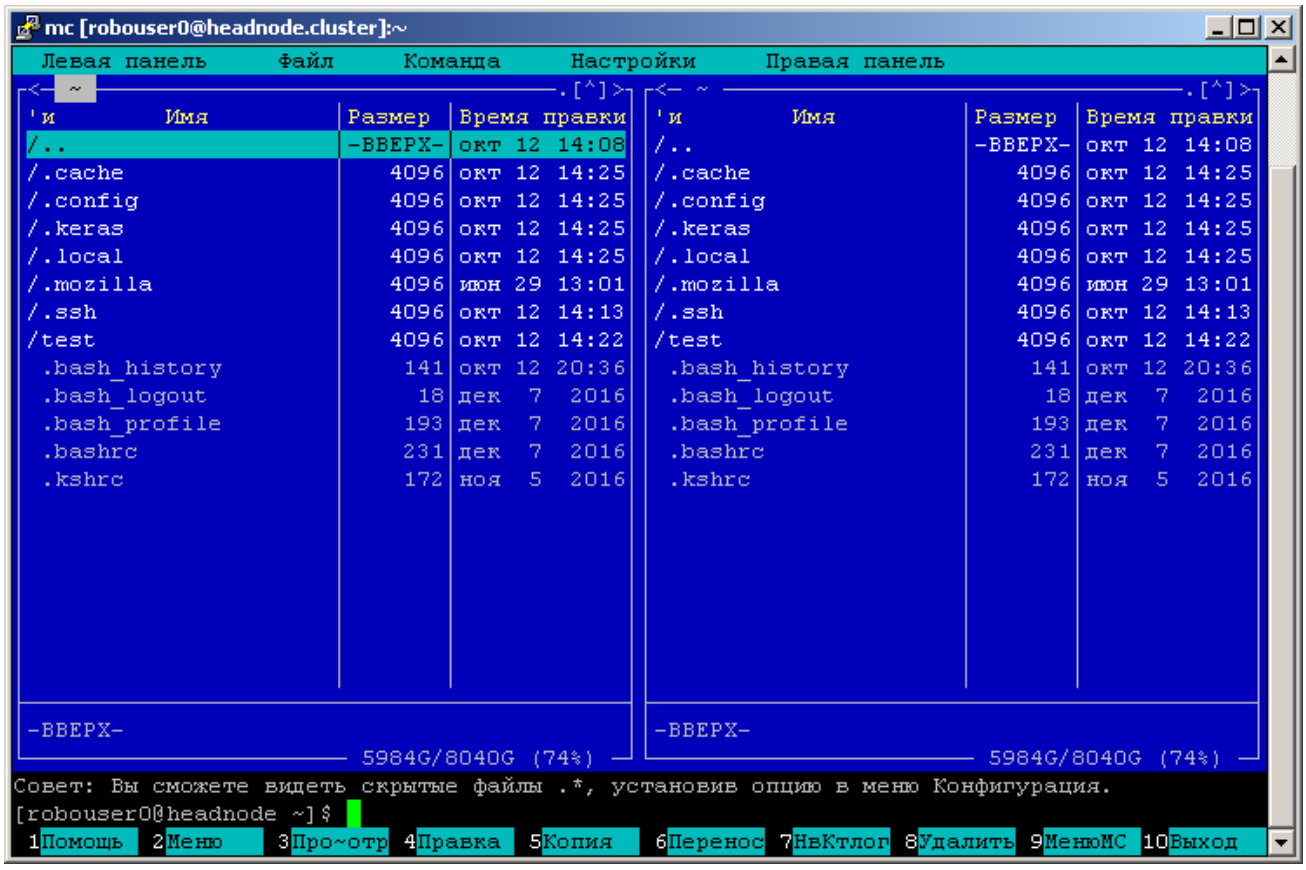
1. Соединитесь с Linux-кластером, используя Putty. Параметры доступа:

hostname: `79.170.167.30`
port: `57322`
usernames: `robouser0, robouser1, ..., robouser9`
(имя пользователя выбирается в соответствии с номером рабочего компьютера)
password: `RoboSchool2017`

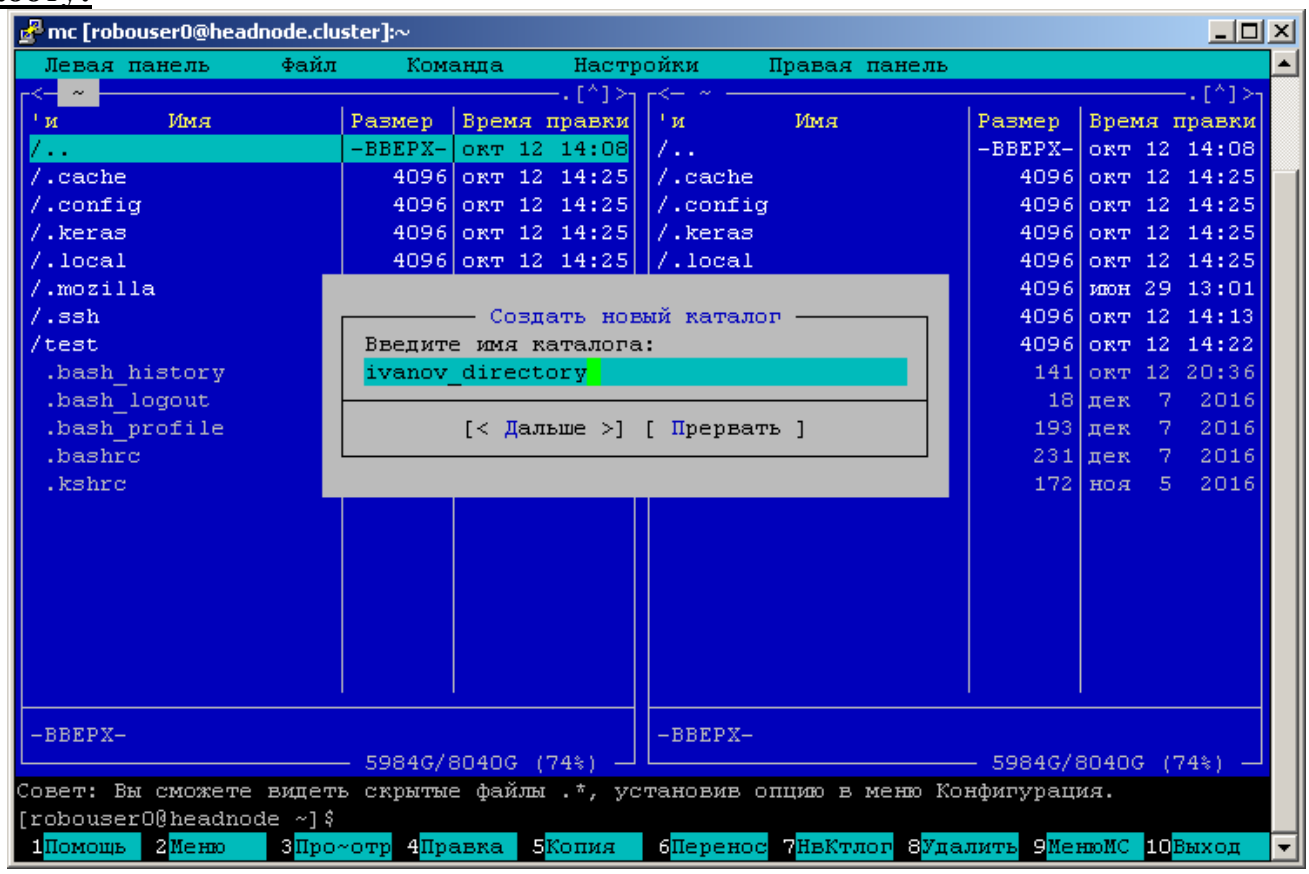


2. Для удобства дальнейшей работы запустите файловый менеджер Midnight Commander:

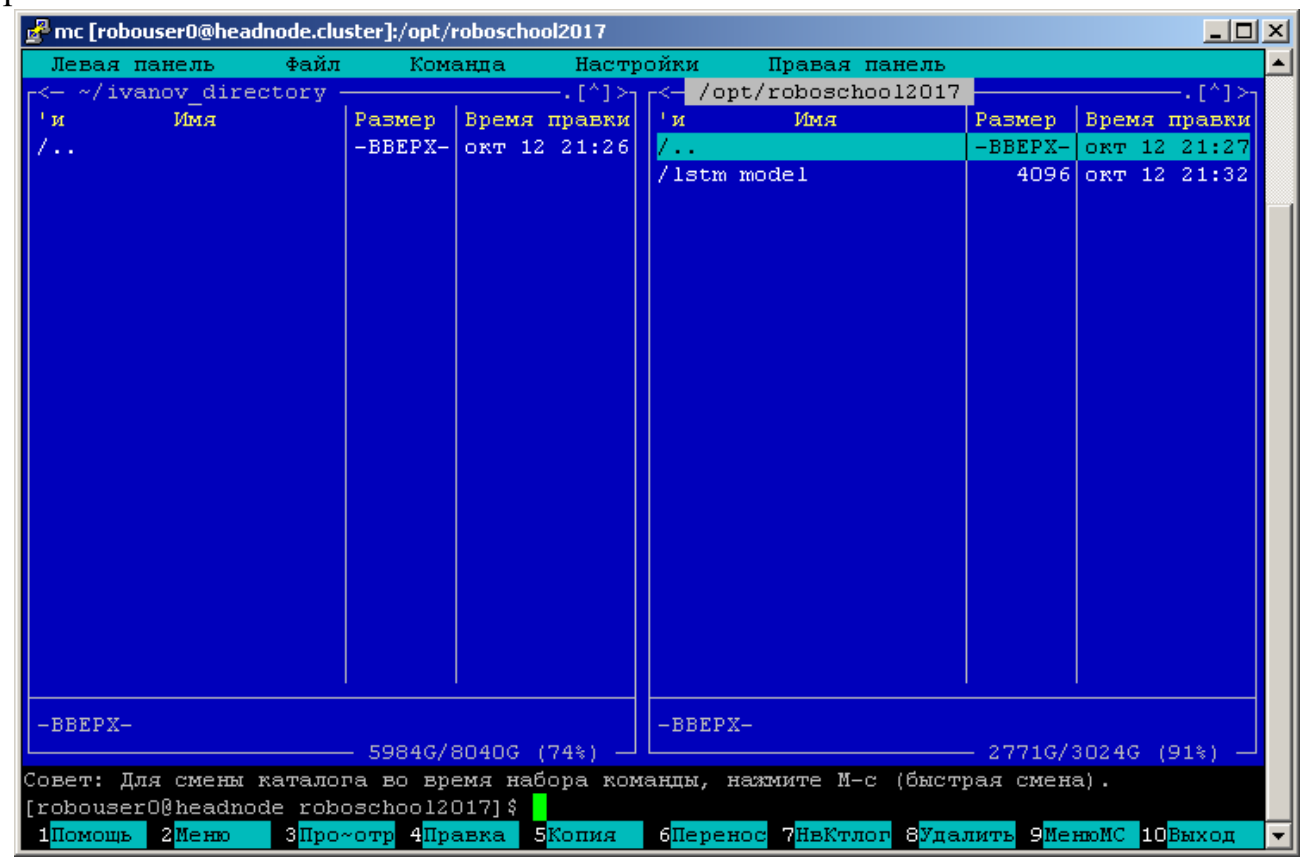
```
[robouser0@headnode ~]$ mc
```



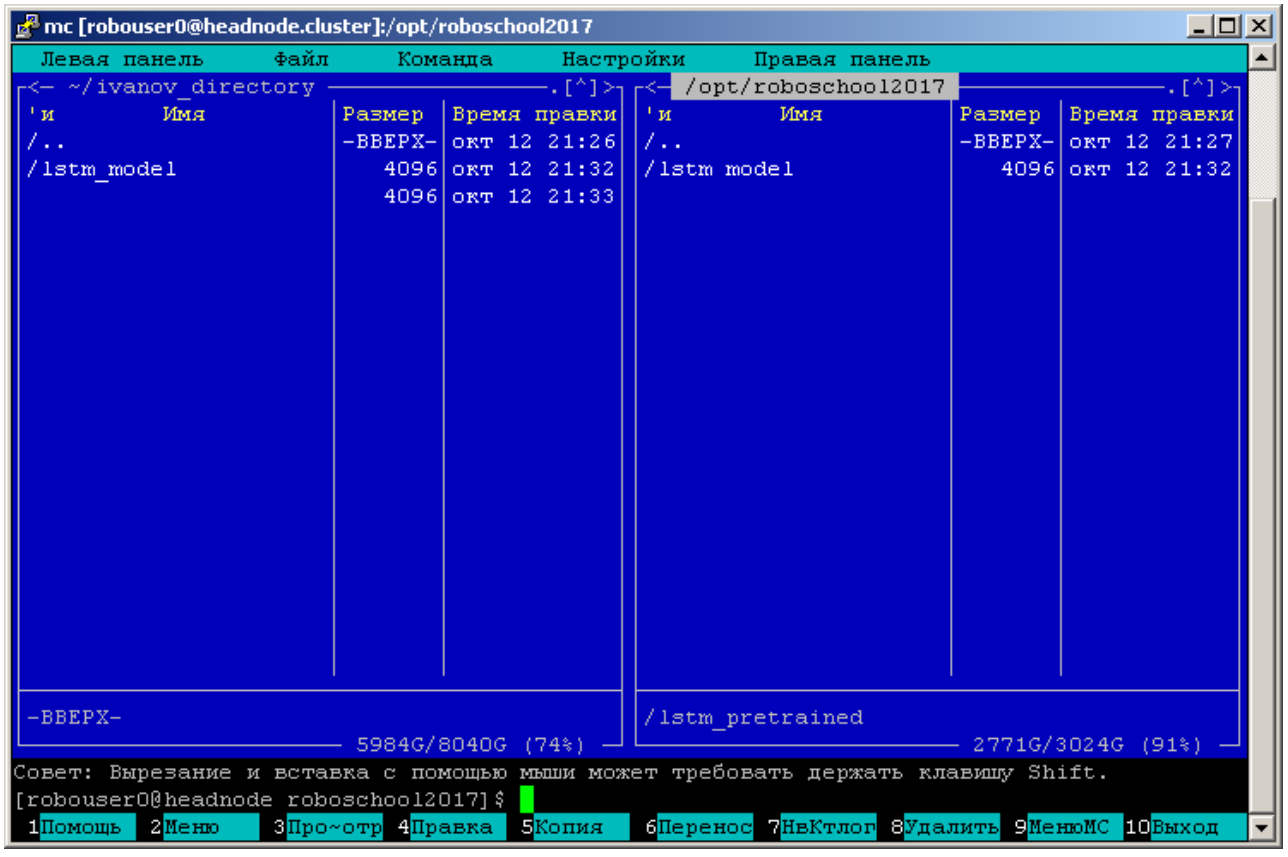
3. Файловый менеджер покажет содержимое домашней директории пользователя. Создайте в этой директории папку, в которой будут располагаться исследуемые файлы и программы, нажатием клавиши **F7**. **Директорию настоятельно рекомендуется называть по имени пользователей, выполняющих работу!**



4. Откройте в левом окне файлового менеджера только что созданную папку, а в правом — директорию `/opt/roboschool2017`. Между окнами менеджера можно переключаться клавишей `Tab`.



5. Скопируйте содержимое папки `/opt/roboschool2017` в рабочую директорию с помощью клавиши **F5**.

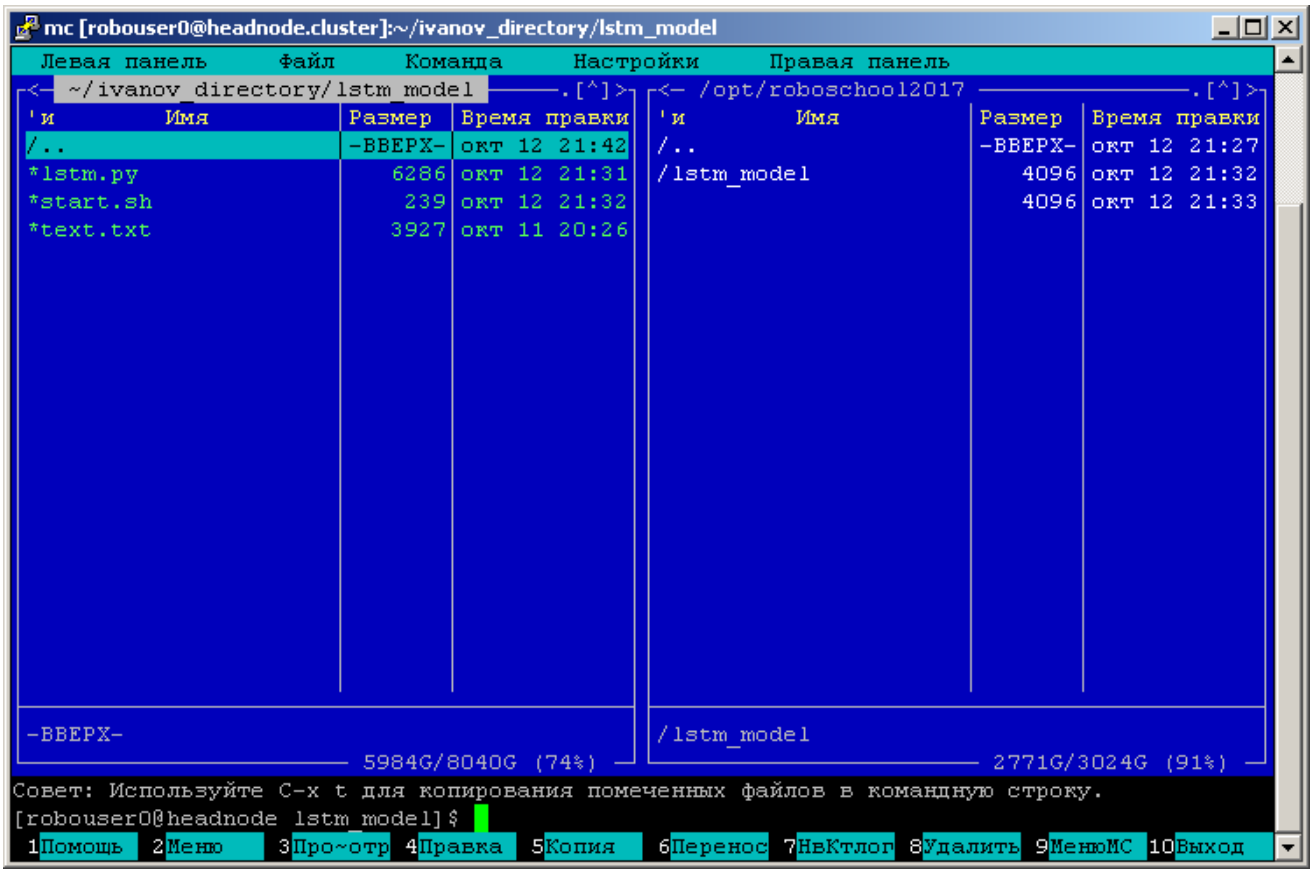


6. Откройте папку `lstm_model` в рабочей директории. В ней расположены три файла:

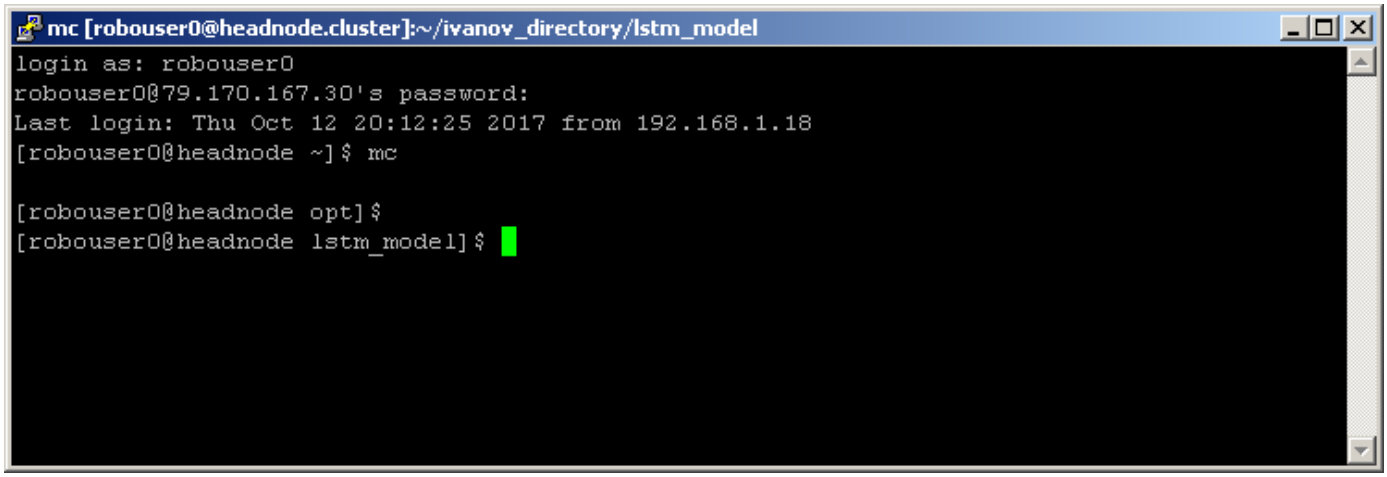
а) `lstm.py` — скрипт на языке Python, выполняющий обучение нейросети и получение предсказаний на его основе;

б) `start.sh` — скрипт запуска задачи обучения нейросети в системе ояередей кластера Slurm;

в) `text.txt` — фрагмент произведения А.С. Пушкина «Евгений Онегин» (вообще говоря, текст может быть любым), на словах которого обучается нейросеть.

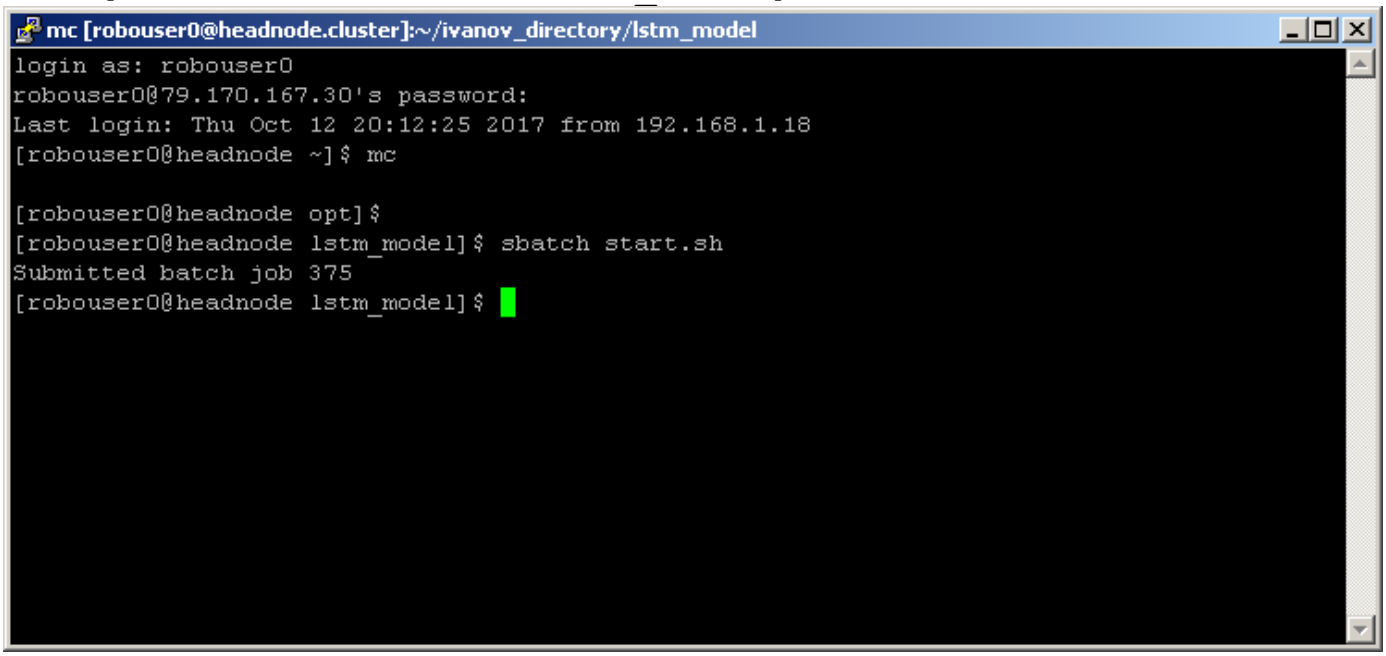


7. Переключите файловый менеджер в режим консольного ввода-вывода, нажав комбинацию клавиш `Ctrl-O` (если потребуется вернуться в оконный режим менеджера обратно, следует нажать `Ctrl-O` повторно).



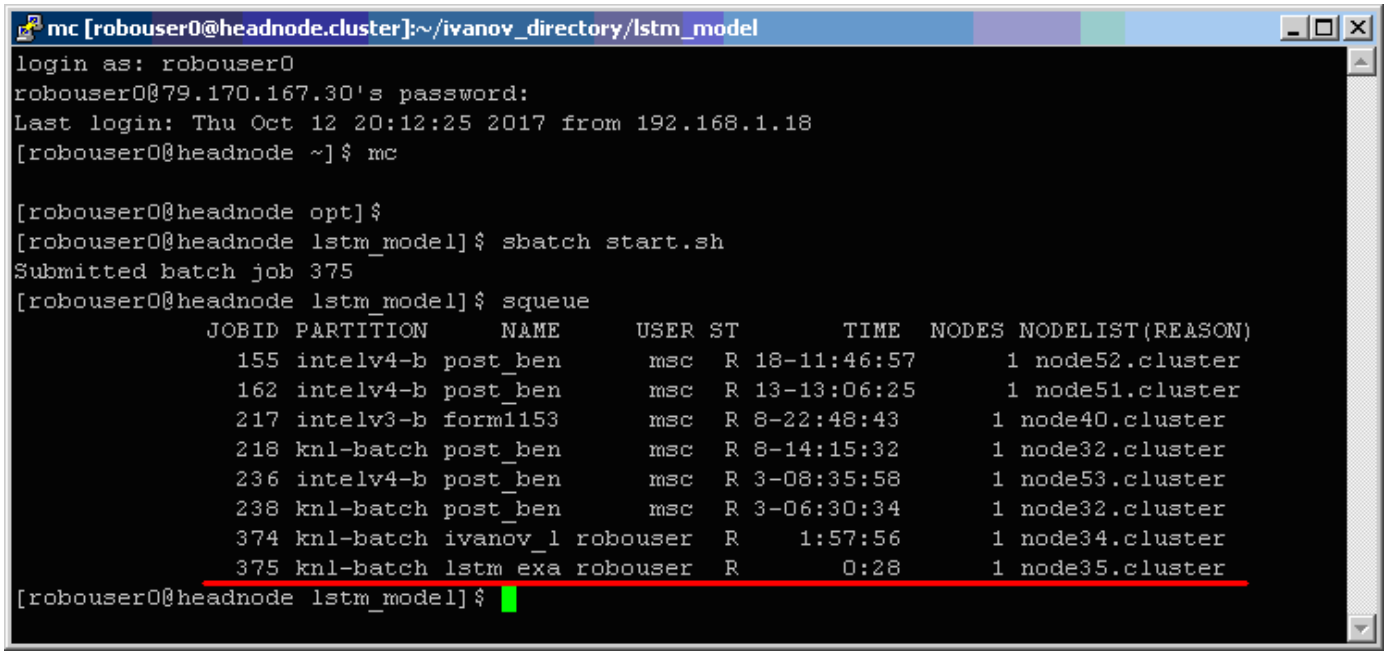
8. Поставьте задачу обучения нейросети в очередь командой:

```
[robouser0@headnode lstm_model]$ sbatch start.sh
```

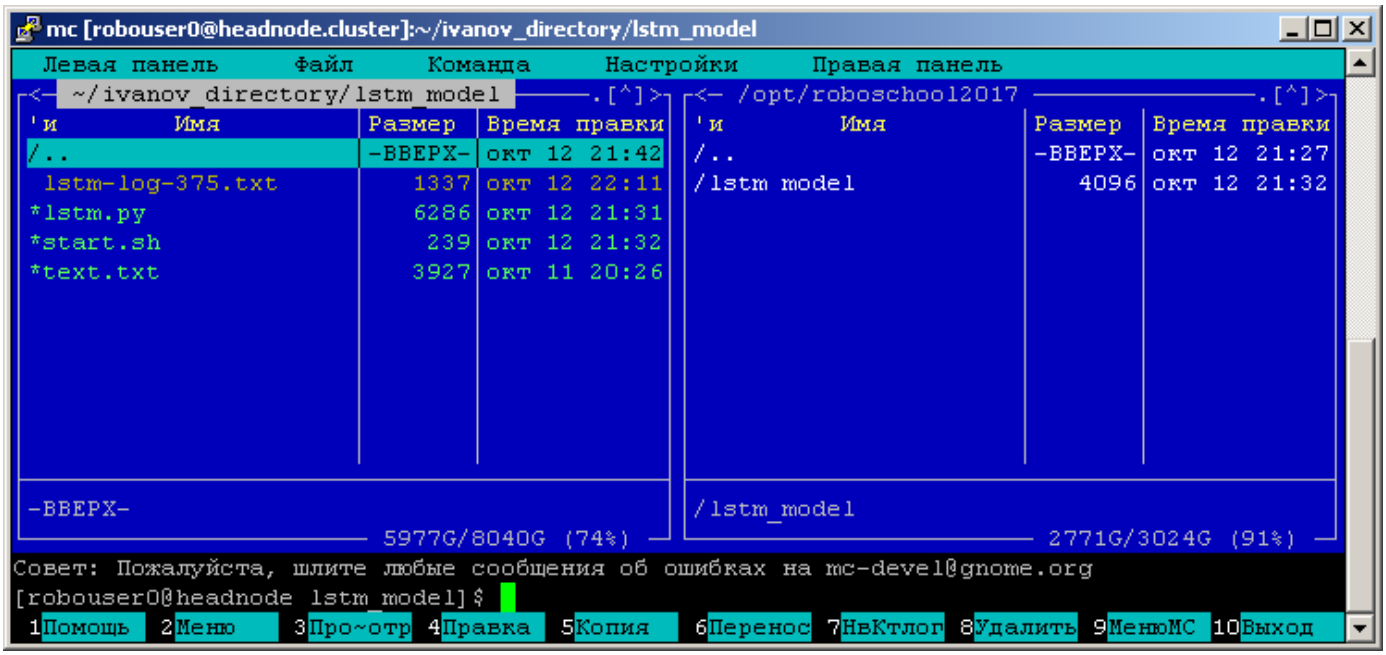


9. После того, как задача поставлена в очередь, ее статус можно посмотреть командой:

```
[robouser0@headnode lstm_model]$ squeue
```

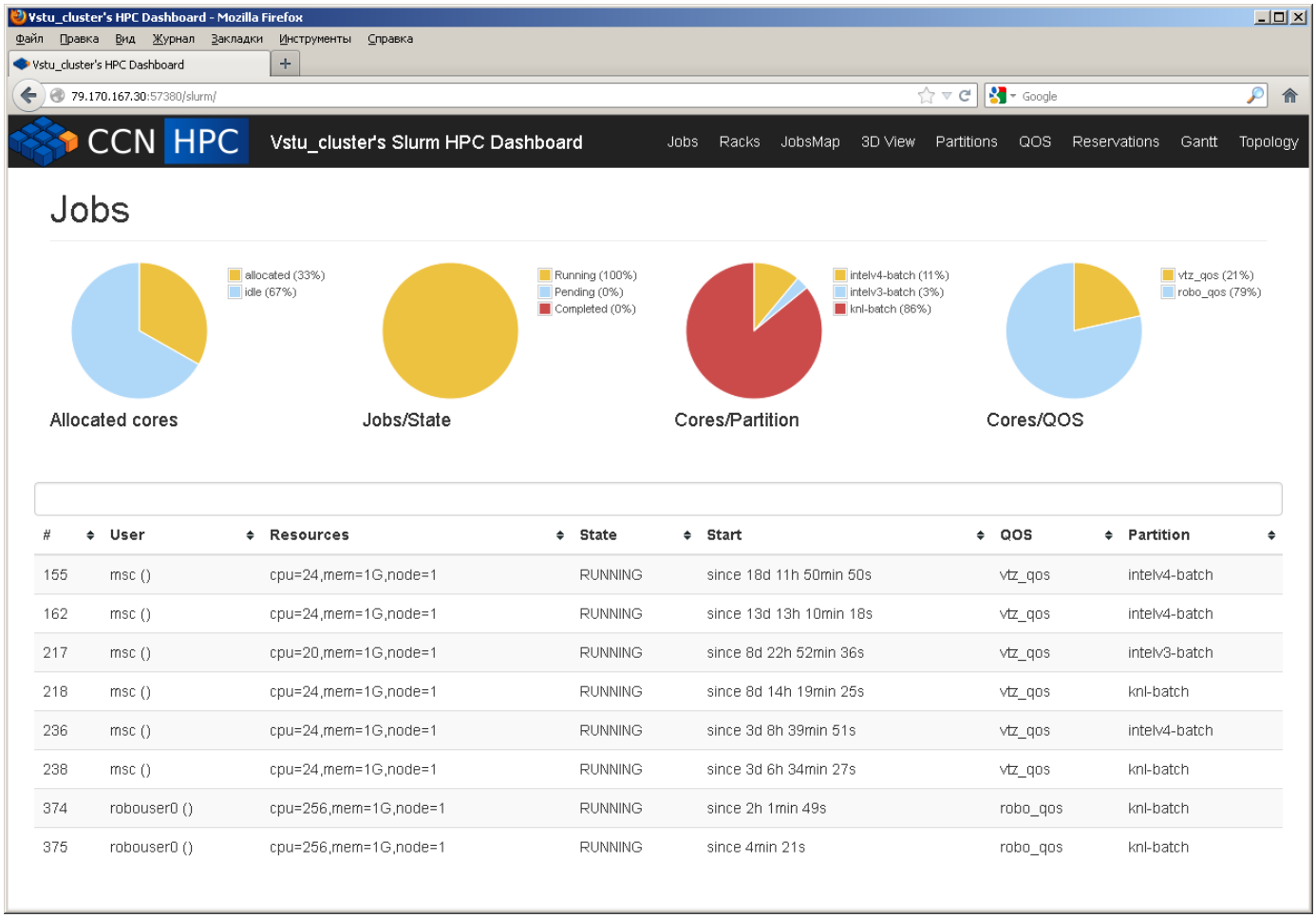


10. После запуска программы в системе очередей на исполнение в директории `lstm_model` появится лог-файл, в который будет записываться вывод программы. При необходимости, его содержимое можно просмотреть, нажав клавишу **F3** (выйти из режима просмотра текста можно с помощью клавиши **esc**).



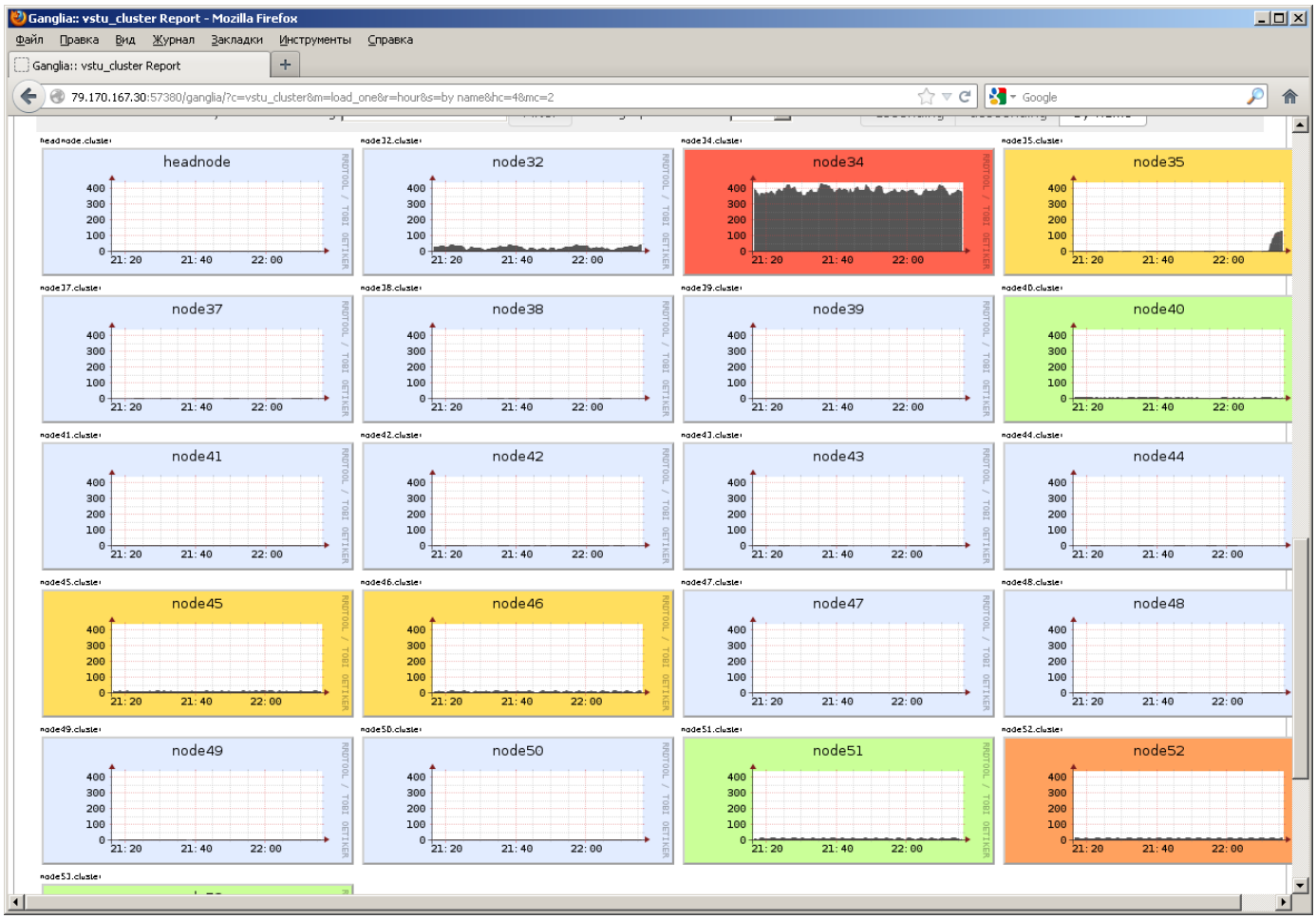
11. Статус задачи в системе очередей можно узнать также с использованием web-интерфейса, перейдя в браузере по адресу:

http://79.170.167.30:57380/slurm

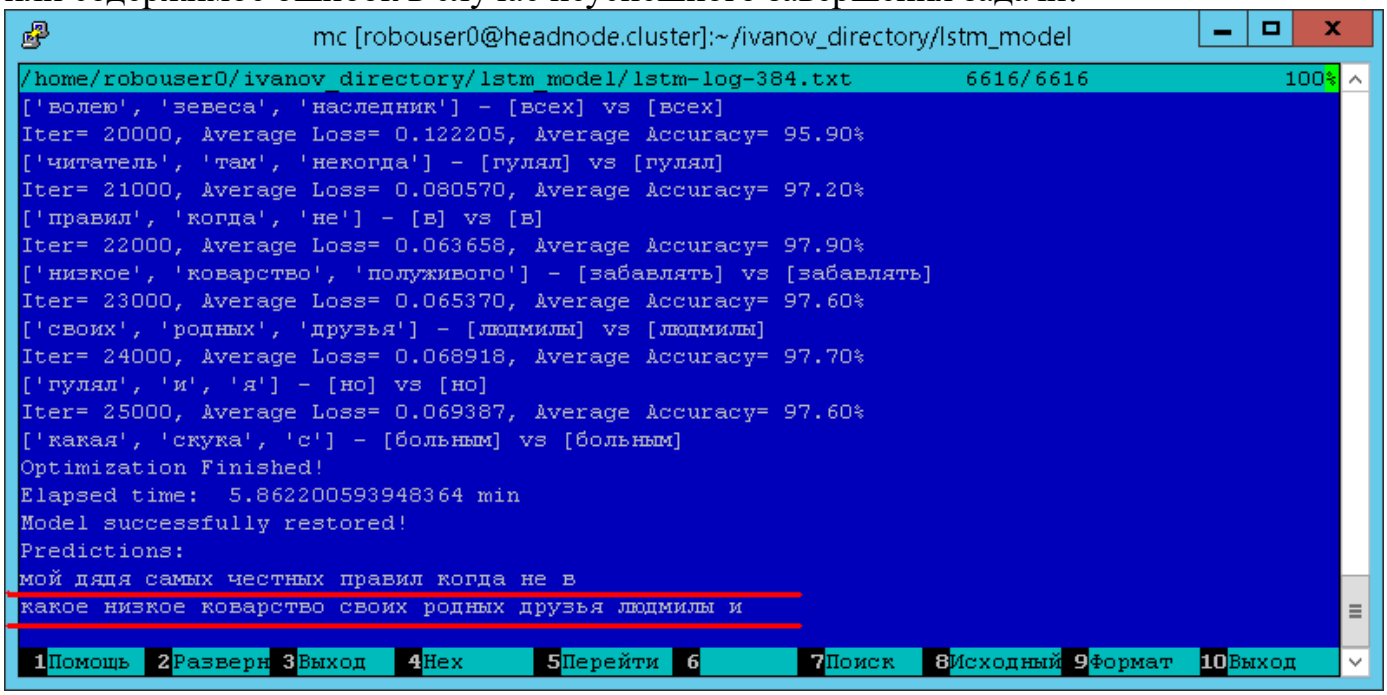


12. Текущую загрузку вычислительных узлов можно посмотреть с помощью системы мониторинга Ganglia, перейдя в браузере по адресу:

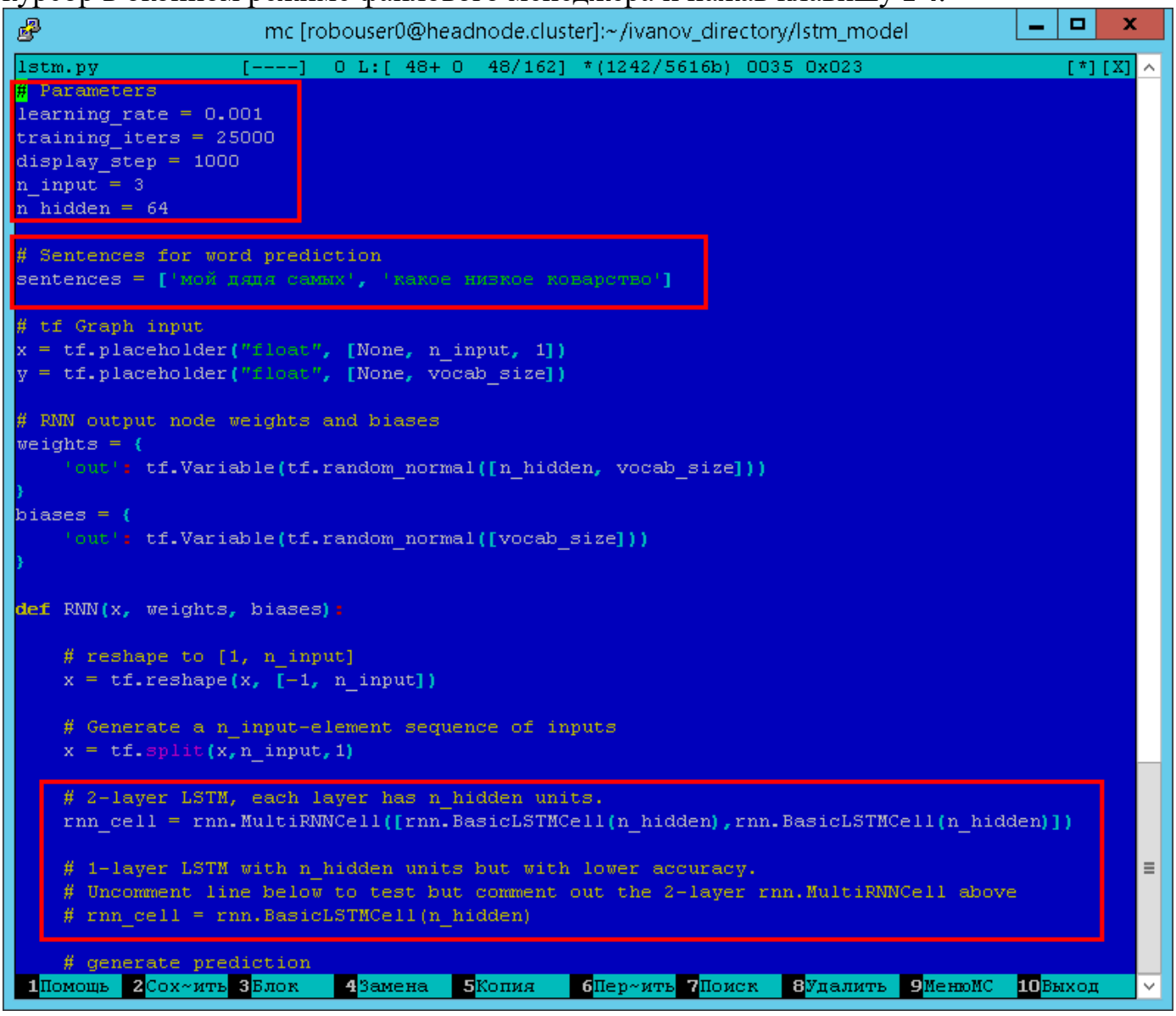
http://79.170.167.30:57380/ganglia



13. После того, как обучение модели будет завершено, начнется процесс предсказания последовательности слов. В качестве исходной последовательности берутся строки из трех слов, заданные в переменной `sentences` в файле `lstm.py`. После завершения работы программы (это можно узнать по статусу соответствующей ей задачи в slurm) результат предсказания можно посмотреть в лог-файле. Содержимое лог- файла позволяет определить расчетное время и точность обучения или содержимое ошибок в случае неуспешного завершения задачи:



17. Результат предсказания модели зависит от длины текста, параметров модели, числа слоев нейросети и т.д. Видно, что результат предсказания неточен, поскольку для ускорения обучения нейросети была использована слишком грубая модель. Для уточнения параметров модели можно поправить файл `lstm.py`, наведя на него курсор в оконном режиме файлового менеджера и нажав клавишу **F4**.



19. После правки параметров удалите файлы сохраненной модели и запустите обучение вновь. Имейте в виду, что чем точнее модель, тем дольше она будет обучаться. Можно взять другой текст для обучения. После получения обученной модели проведите предсказание слов и сравните качество предсказания с предыдущим примером. Сделайте выводы.