Adam Mickiewicz University in Poznan

Doctoral School of Exact Sciences AMU

Linux Shell Programming

Prof. UAM dr hab. Grzegorz Musiał

Scientific lectures, workshops

Field of science	Exact and natural sciences
Teaching method	Computer lab with lecture elements
Language	English
Numbers of hours	15
Aims of the course	Basics and practical skills of working in the Linux operating system, including remote computers, protection of user resources, and file transfer between computers. Automate a large part of the scientific work with scripts, including searching for data from many (even tens or hundreds) files, as well as storing this data in files to create graphs. Practical skills in running computational tasks in a queuing system on high-performance computers.
Course contents	1. The role and functionalities of Linux operating system, advantages of this operating system over others, protection of user resources. 2. Working in Linux operating system, taking advantage of its key functionalities, including work on a remote computer and transferring of files. 3. Shell programming with execution of a large number of sample scripts. 4. PhD candidates create own scripts for automation of their scientific work, including searching for data from many (even tens or hundreds) files, as well as storing this data in files to create graphs. 5. Running computational tasks in a queuing system on high-performance computers. 6. Using information sources in the Linux operating system, bibliographic and Internet sources to independently solve problems with automatic collection of large amounts of data in many files.
Prerequisites and co-requisites	Basic knowledge and skills of using the UNIX/Linux operating system. Nevertheless, in the absence of them, it is possible to introduce students to these basics at the expense of a slight reduction in the program.

Learning outcomes	
On completion of the course PhD candidates will be able to:	**Assessment mode**
explain the role and functionalities of Linux operating system, advantages of this operating system over others, protection of user resources (E_W01, E_W02, E_U06, E_K02, E_K05)	Individually solved and assessed tasks performed in the computer laboratory with explanation to other students
work in Linux operating system, taking advantage of its key functionalities, including work on a remote computer and transferring of files as well as run computational tasks in a queuing system on high-performance computers (E_W02, E_U06, E_K05)	Individually solved and assessed tasks performed in the computer laboratory with explanation to other students

create and present own scripts for automation of her/his scientific work, including searching for data from many (even tens or hundreds) files, as well as storing this data in files to create graphs (E_W01, E_W02, E_U06, E_K05)	Individually solved and assessed tasks with explanation to other students
benefit from system information sources, as well as from bibliography and internet sources for independent solving problems (E_U02, E_K05)	Individually solved and assessed tasks

Literature	[1] D. Tansley, Linux & Unix Shell Programming, Addison-Wesley, 2000, ISBN13: 9780201674729 M. Garrels, Introduction to Linux, ver. 1.9 20040101, the GNU Free Documentation License, http://www2.pv.infn.it/sc/linux/Intro-Linux.pdf [2] K. Haviland, D. Gray, B. Salama, Unix System Programming, Addison-Wesley, 1999, ISBN-13: 978-0201877588; in Polish: K. Haviland, D. Gray, B. Salama, Unix: programowanie systemowe, Wydawnictwo RM, Warszawa 1999, ISBN 83-7243-016-0 [3] K. A. Robbins, Steven Robbins, Unix™ Systems Programming: Communication, Concurrency, and Threads, Prentice Hall PTR, 2003, ISBN 0-13-042411-0 https://cs.uwec.edu/~tan/priv/www-docs/cs462/USP.pdf
Additional information	Available to PhD students, we have excellent facilities for these classes in the form of a Linux computer lab with the latest computers and a remotely accessible cluster with several nodes with many installed environments and a queuing system. Thus, PhD students will be prepared theoretically and practically to automate a large part of their scientific work and to run tasks on smaller or larger remote resources, such as computer clusters, high-performance computers or computer grids.