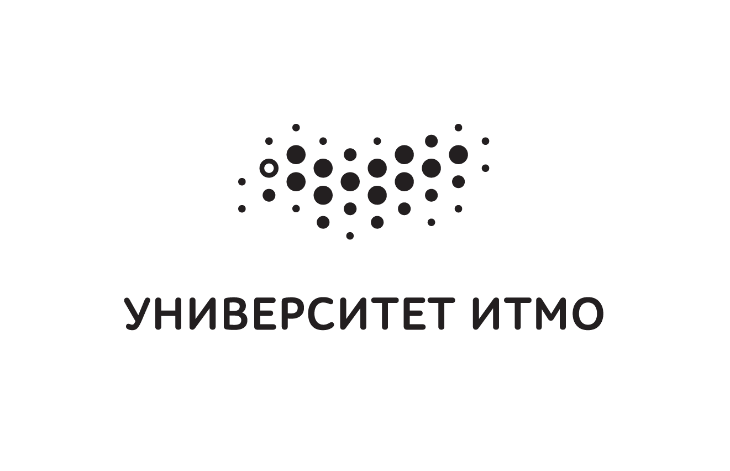
Санкт-Петербургский Национальный Исследовательский Университет Информационных Технологий, Механики и Оптики

ФКТиУ, кафедра Вычислительной техники

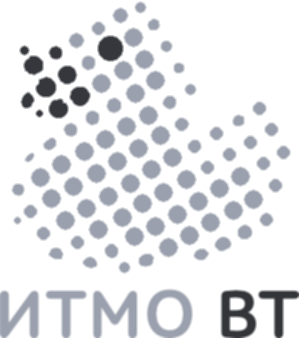


Лабораторная работа №1

по дисциплине

«Системное программное обеспечение»

Выполнил: Студент группы P3202



Преподаватель: доцент кафедры ВТ

Дергачев Андрей Михайлович

Санкт-Петербург

2018 г.

**Выполнение**

* Список команд, подлежащих повторению:

ls, pwd, cd, rm, mv, cp, mkdir, rmdir, type, file, find, chmod, ln, wc, tee, cat, tail, head, more, pg, touch

|  |
| --- |
| 1. **ls (LiSt) [-options] [files]** |
| For each file that is a directory, ls lists the contents of the directory. For each file that is a regular file, ls repeats its name and any other requested information. The output is sorted alphabetically by default. When no arguments are given, the current directory is listed. |
| **Basic keys**  -a lists all entries, including those that begin with a dot, which are normally not listed  -l list in long format,giving mode , number of links, owner, group, size in bytes, and time of last modification for each file ,..  -i for each file, prints the i-node number in the first column of the report |
| **Environment Variables:** LANG, COLUMNS, LC\_ALL, LC\_MESSAGES, NLSPATH, TZ |
| **Is the filter:** no |

|  |
| --- |
| 1. **pwd (Print Working Directory)** |
| The pwd utility writes an absolute path name of the current working directory to standard output |
| **Is the filter: no** |

|  |
| --- |
| 1. **cd (Change working Directory) [directory]** |
| Change the current directory to the showed direc |
| **Basic keys:**  -L use the physical directory structure  -P follow sympolic links |
| **Environment variables:** HOME, PWD, OLDPWD |
| **Is the filter:** no |

|  |
| --- |
| 1. **rm (ReMove) [-options] [files]** |
| Remove showed object from system ( file, directory, link) |
| **Basic keys:**  -r, -R recursively delete directories and subdirectories  -f don“t print messages about error  -i request confirmation when deleting a file |
| **Is the filter:** no |

|  |
| --- |
| 1. **mv (MoVe) [-options] [file1] [file2]** |
| Move the files |
| **Basic keys:**  -f don“t print messages about error  -i mv will prompt for confirmation whenever the move  would overwrite an existing target. An affirmative  answer means that the move should proceed. Any  other answer prevents mv from overwriting the tar-  get. |
| **Is the filter:** no |

|  |
| --- |
| 1. **cp (CoPy) [-options] [file1] [file2]** |
| Copy file1 to file2 |
| **Basic keys:**  -r, R copy recursive  -f allow deleting the file to be copied if it cannot be opened for writing  -i request confirmation when copying  -p copy date last changes, group id, permissions of the source file  -n do not overwrite existing file |
| **Is the filter:** no |

|  |
| --- |
| 1. **mkdir (MaKe a DIRectory) [-options] [files]** |
| Create directories with the name specified in the argument (if they do not already exist). |
| **Basic keys:**  -m override default permissions  -p before creating the target directory, creates all the parent (if those do not exist) |
| **Is the filter:** no |

|  |
| --- |
| 1. **rmdir (ReMove a DIRectory) [-options] [files]** |
| Deletes directories, operand should be an appeal to an empty directory |
| **Basic keys:**  -p delete also parent |
| **Is the filter:** no |

|  |
| --- |
| 1. **type (Write a description of command TYPE) [-options] [names]** |
| Specify how each argument will be interpreted if it is used as the command name. |
| **Enviroment variables:** PATH |
| **Is the filter:** no |

|  |
| --- |
| 1. **file (Determaine FILE type) [-options] [files]** |
| Specifies the type of file |
| **Basic keys :**  -c check file for format errors  -f file contains a list of files to be checked |
| **Is the filter:** no |

|  |
| --- |
| 1. **find (FIND) [path] [-options]** |
| Search files by names, properties and other atributes |
| **Keys :**  -name by username;  -typ by type of file;  -user by user;  -group by group;  -perm by permissions |
| **Is the filter:** no |

|  |
| --- |
| 1. **chmod (CHange the permissions MODe of a file) [-options] [files]** |
| Change permissions |
| **Basic keys :**  -R recursively change permissions for directories and their contents  -f don“t print messages about error  -v describe in detail the action for each file |
| **Is the filter: no** |

|  |
| --- |
| 1. **ln (make links between files) [-option] [file1] [file2]** |
| If there is no key, create a hard link to the file.  With the -s option, create a symbolic link to the file. |
| **Basic key :**  -s create symbolic link |
|  |
| **Is the filter : no** |

|  |
| --- |
| 1. **wc (Word Count) [-options] [files]** |
| Print the number of line, word and byte translations for each specified file, and the total count if multiple files were specified. If the input file is not specified, or is - , then the data is read from the standard |
| **Basic keys :**  -l number of lines  -w number of words  -m number of symbol  -c number of bytes |
| **Is the filter:** yes |
| **Example: $** echo 'Hello World!!!' | wc -lwmc  **1 2 15 15** |

|  |
| --- |
| 1. **tee [-options] [files]** |
| Named after the T-spliter in the aqueduc, which divides the water into two streams and is shaped like the capital T |
| Read from standard input stream and redirect to standard ouput stream and files specified in argument |
| **Basic keys :**    -a add output data to the end of the file without overwritting its content  -i ignore error |
| **Is the filter:** yes |
| **Example: $** echo “GGWP” | tee file1 file2 |

|  |
| --- |
| 1. **cat (conCATinate) [-options] [files]** |
| Displays sequentially specified files, combining them into a single stream, if ’-’ - is indicated instead of a file, it reads the standard input stream |
| **Basic keys:**  -n before each line of output write its number  -v display unprintable characters |
| **Is the filter:** yes |
| **Example:** **$** echo “GGWP” | cat  “GGWP” |

|  |
| --- |
| 1. **tail [-options] [files]** |
| Display n ( 10 by default) last line of file |
| **Basic keys :**  -n print n lines  -r print n lines in reverse |
| **Is the filter:** yes |
| **Example:** ls | head -2  **test1**  **test2** |

|  |
| --- |
| 1. **head [-options] [files]** |
| Display n ( 10 by default) first line of file |
| **Basic keys :**  -n print n lines  -r print n lines in reverse |
| **Is the filter:** yes |
| **Example:** ls | tail -2  **test9**  **test10** |

|  |
| --- |
| 1. **more [-options] [file]** |
| View the contents of the file with a stop at the border of the screen |
| **Basic keys :**  -c clear the screen before viewing  -s replace multiple blank lines with one  -w waiting for the key to be pressed before existing |
|  |
| **Is the filter:** yes |
| **Example:** ls -l | more -c |

|  |
| --- |
| 1. **pg (PaGe) [-options] [file]** |
| Displays file contents page by page. |
| **Basic keys :**  -с clear the screen before displaying each page |
| **Is the filter:** yes |
| **Example:** ls -l | pg -c |

|  |
| --- |
| 1. **touch [-options] [files]** |
| Overwrite the time of access and last change to the current fate  Can be used to create empty file |
| **Basic keys :**  -a change access time only  -m change only the modification time  -c don“t create non-existent files |
|  |
| **Is the filter:** no |

* Список команд, подлежащих повторению:

su, chown, chgrp, less, split, join, paste, cut, tr, cmp, diff, patch, sort, uniq, echo, alias, ulimit, umask, groups, id, getent, xargs

|  |
| --- |
| **22) su (Substitute User, Super User или Switch User) - [username]** |
| Allows the user to log in with a different name,not end the current session |
| **Basic key:**  -c execute one command  -l, - work on behalf of the user in his initial enviroment |
| **Environment variables:**  LC\_CTYPE how command handles characters  LC\_MESSAGES determines how diagnostic and informative messages are presented. |
| **Is the filter? :** no |
| **Example: $ su — bin — c „command args“** |

|  |
| --- |
| **23) chown (CHange file OWNership) [-options] [owner:group] [files]** |
| Change owner/group for everyfile to specified in owner |
| **Basic key :**  -f don“t show messenger about errors  -h change the owner of the symbolic link  -R change recursively  -c discribe in detail the action for each changing file |
| **Is the filter:**  no |
| **Example: $ chown user:somegroup /hello /hellofile** |

|  |
| --- |
| **24) chgrp (CHange file GRouP owner) [-options] [group] [files]** |
| Change group of the file |
| **Basic key**  Like command „chown“ |
| **Is the filter:** no |
| **Example: $ chgrp -R testgroup /testdir** |

|  |
| --- |
| **25) less [file]** |
| Display a scrollable file |
| **Environment variables:** HOME(Uses s home directory), COLUMNS |
| **Is the filter :** yes |
| **Example: $ cat /file1 /file2 | less** |

|  |
| --- |
| **26) split [-options] [file] [name]** |
| Copy the file and divide it into files according to the specified keys.  The output file names will be composed of this prefix and two additional letters aa, ac,ac, etc. If thr prefix of file names is not specified, then x is used by default |
| **Basic key:**  -l number of lines  -b [n] number of bytes  -b [n]k number of bytes \*1024 |
| **Is the filter:** no |
| **Example: $ split -l2 -d /testfile newfile;**  **ls newfile00 newfile01 newfile02** |

|  |
| --- |
| **27) join [-options] [file1] [file2]** |
| Combines the lines of two orderd text files based on the presence of a common feild |
| **Is the filter:** yes |
| **Example:**  **$ echo “first text1-1\nsecond text1-2” > file1**  **$ echo “first text2-1\nsecond text2-2” > file2**  **$ join file1 file2 > file3; cat file3**  **first text1-1 text2-1**  **second text2-1 text2-2** |

|  |
| --- |
| **28) paste [-s] [files]** |
| View files as vertical column, connect them and output the result to standard output |
| **Basic key :**  -s changes the position of rows with columns  -d change the delimiter (as TAB) |
| **Is the filter:** yes |
| **Example: $ paste test1 test2 > test3; cat test3**  **col1 col2 col3 col4**  **text1 text2 text3 text4** |

|  |
| --- |
| **29) cut [-options] list [files]** |
| Retrieves ranges of bytes, characters, columns |
| **Basic key:**  -b select bytes at a given position  -c select only characters in given position  (position or : -4,1-7,10-)  -f display only the columns listed in the list(column number)  -d change delimiter(default is TAB) |
| **Is the filter:** yes |
| **Example: cat test3 | cut -f2-3 -d' '**  **col2 col3**  **text2 text3** |

|  |
| --- |
| **30) tr (TRanslate or delete characters) [-options] [str1] [str2]** |
| Replace or delete selected characters. Tr takes no arguments, the working line is taken from standard input or from thr output of other programs |
| **Basic keys:**  -d delete all characters specified in str1  -s replace each character specified in str1 with that soecified in str2  in case of specifying several characters in str1 and str2 replacement  produced in the order of conformity(tr 12 ab, 1 in a, 2-b) |
| **Is the filter:** да |
| **Example: $ echo “abcdefgh” | tr “a-d” “e-h” | cat**  **“efghefgh”** |

|  |
| --- |
| **31) cmp (CoMPare two files byte by byte) [-options] [files] [skip]** |
| Compares two files and if they are different, reports the first byte and the line where the difference was found |
| **Basic keys:**  -l displaying the differing bytes and their numbers  -s output nothing but the result status for differentfiles |
| **Is the filter:** yes |
| **Example: $ echo “test2” | cmp test1**  **test1 - differ: byte 1, line 1** |

|  |
| --- |
| **32) diff (DIFFerence) [-options] [file1] [file2]** |
| Compare line by line files(need to be changed in file 1 to get file2)  a — add lines, c — change lines, d — delete lines |
| **Basic keys:**  -w ignore tabs and spaces  -i ignore case  -s output the report if the files are identical |
| **Is the filter:** yes |
| **Excample: $ echo “a b c\nd e” > diff1; echo “a b c” > diff2**  **$ diff diff1 diff2**  **2d1**  **< d e** |

|  |
| --- |
| **33) patch [-options] < [file]** |
| Transfer edits between different versions of text files |
| **Basic keys:**  -b saves a copy of thr original contents of files  -R undo changes |
| **Is the filter:** no |
| **Example: $ diff diff1 diff2 > diff.patch**  **$ patch diff2 < diff.patch**  **a b c** |

|  |
| --- |
| **34) sort [-options] [files]** |
| Sort line by line all the files specified in the argument. Outputto standard output |
| **Basic keys:**  -f ignore case  -r reverse order |
| **Environment variables:** LC\_NUMERIC (fraction display) |
| **Is the filter:** yes |
| **Example: echo “b \nc \nn \nw \na \nx” | sort**  **a**  **b**  **c**  **n**  **w**  **x** |

|  |
| --- |
| **35) uniq (UNIQue) [-options] [in\_file] [out\_file]** |
| Filter duplicate lines from a sorted in\_file file and display them and related information in out\_file |
| **Basic keys:**  -c print the number of repetitions  -d only duplicate lines  -u only unique lines |
| **Is the filter** no |
| **Example: $ echo “1 \n1 \n2 \n2 \n2 \n3” | uniq -c**  **2 1**  **3 2**  **1 3** |

|  |
| --- |
| **36) echo [string]** |
| Print text to standard output |
| **Is the filter:** no |
| **Example: $ echo “hi, teacher!”** |

|  |
| --- |
| **37) alias [name[=value]]** |
| Create an alias(shorter record) for a command or pipeline |
| **Is the filter:** no |
| **Example: $ alias list5="ls -l | head -n 5"** |

|  |
| --- |
| **38) ulimit (User LIMIT) [-options] or -a** |
| Establish (or simply deduce) system resource limits |
| **Basic keys:**  -a remove all restrictions  -d maximim data segment size  -u maximum number of processes run by this user  -n maximum number of open files  -f maximum file size  -v maximum size of used virtual memory |
| **Is the filter:** no |
| **Example: $ ulimit -a**  **virtual memory (kbytes, -v) unlimited**  **max memory size (kbytes, -m) unlimited**  **…** |

|  |
| --- |
| **39) umask (User MASK) [-s] [mask]** |
| Specify which permission bits to reset when creating subsequent files |
| **Is the filter:** no |
| **Example: $ touch test\_file1**  **$ umask 444**  **$ touch test\_file2**  **$ ls -l**  **-rw-rw-rw- 1 ductran ductran 0 Mar 3 16:00 test\_file1**  **--w--w--w- 1 ductran ductran 0 Mar 3 16:01 test\_file2** |

|  |
| --- |
| **40) groups [-username]** |
| Lists groups for specified users or current process |
| **Is the filter:** no |
| **example: $ groups**  **studs p3202** |

|  |
| --- |
| **41) id (real and effective user and group IDs) [-options] [-user]** |
| |Information about the specified user or (if without parameters) about the user running the program |
| **Is the filter:** no |
| **Example: id**  **uid=247411(s247411), gid=600(studs)** |

|  |
| --- |
| **42) getent (GET ENTries) [-options] database [key]** |
| Displays a list of records from the database, supported libraries Name service Switch |
| **Is the filter:** no |
| **Example: getent passwd ductran** |

|  |
| --- |
| **43) xargs (build and execute command lines)**  **[-*options*] [*command* [*initial-arguments*]]** |
| Combines a fixed set of initial arguments specified on the command line with arguments read from the standart input, and executes the specified command one or more times |
| **Basic keys:**  l execute a command for each group of a specified numnber of non-empty argument lines read from standard input  -t tracing mode;  -p invitation mode;  -n execute the command using the maximum possible number of arguments read from the standard input, but nomore than the specified number |
| **Is the filter:** no |
| **Example: echo first second third | xargs -n 1 echo**  **first**  **second**  **third** |

**Описание атрибутов и интерпретация прав доступа**

**$ ls -l**

drwxr-xr-x 2 s247411 studs 2 июня 18 2018 Desktop

drwxr-xr-x 2 s247411 studs 2 июня 18 2018 Documents

drwxr-xr-x 3 s247411 studs 3 дек. 24 17:33 glassfish

drwxr-xr-x 6 s247411 studs 7 дек. 24 01:14 glassfish5

drwxr-xr-x 2 s247411 studs 14 июня 18 2018 ku4

drwxr-xr-x 11 s247411 studs 11 дек. 26 17:18 lab2

drwxr-xr-x 2 s247411 studs 6 окт. 10 01:40 lab6\_pr

drwxr-xr-x 2 s247411 studs 6 февр. 19 03:09 lab7

drwxr-xrwx 2 s247411 studs 2 дек. 26 14:55 pip\_2

drwxr-xr-x 2 s247411 studs 2 дек. 26 13:46 pip2

drwxr-xr-x 4 s247411 studs 8 окт. 1 02:12 public\_html

**Описание атрибутов**

1) Type of file

1. - normal file

2. d directory

3. l symbol link

4. c character device(for accessing computer hardware)

5. b block device (random access, example hdd)

6. p (pipe, pipeline) interprocessor channel

7. s socket

2) File permissions ( in symbolic form)

Consistenely for the user, group, all others

r — read, w — write, x — execute

3) number of hardlink

4) owner“s name

5) group of owner

6) size of file(in byte)

7) data of the last change of file

8) name of file(for symbolic link, name of link → name of file to which it refers)

**Access permissions**

u — user, g — group, o — other, a — all

r — read, w — write, x - execute

Access modifiers for files and hard links

r — view

w — write/edit file

x — execute as a program

**Directory access modifiers:**

r — view content

w — create files and directories inside this directory

x — enter the directory

*For sumbolic links, permissions are determined by the rights of the file on which they point*

**Differences in terminology**

*Command* — interpreter action symbol

Microcommand — elementary actions that make up the team

Nanocommand — command that performs the most elementary

*Programm* — general concept, a set of commands for a specific artist

*Utiluty* — program aimed at performing a specific (highly specialized) task

**Дополнительное задание №1**

$ alias msg=”(tail file -n2; head file -n4) | tr ‘\n’ ‘ ‘ >&2”; msg

**Дополнительное задание №2**

$ (head file -n2 | tail -n1; echo 'to THAT ';head file n4 | tail n1;echo ‘--’; tail file -n2) | tr '\n' ' '>&2;

**Выводы**

There are commands that simplify work with structured text files. The use of these commands in some cases can replace the use of text editors.