

Linux administration

Коник Руслан
E-mail: rulyabox@gmail.com

План

- Basics and terminology
- I/O subsystem

Linux history

1969

- 1969 – рік випуску мови C та операційної системи Unix, які розробили Денніс Рітчі та Кен Томпсон.
- Після початку комерційного продажу Unix отримали два варіанти:
 - AT&T Unix;
 - BSD Unix;
- https://en.wikipedia.org/wiki/Dennis_Ritchie
- https://en.wikipedia.org/wiki/Ken_Thompson
- <https://en.wikipedia.org/wiki/BSD>
- [https://en.wikipedia.org/wiki/Comparison of BSD operating systems](https://en.wikipedia.org/wiki/Comparison_of_BSD_operating_systems)

Linux history

1980s

- У 1980х роках багато компаній розробляли свій власний варіант UNIX: IBM – AIX, Sun – SunOS, HP – HP-UX і т.д.
- Для того, щоб припинити безладну діалектику UNIX Річард Столлман запустив проект GNU.
- https://en.wikipedia.org/wiki/Richard_Stallman
- https://en.wikipedia.org/wiki/IBM_AIX
- <https://en.wikipedia.org/wiki/HP-UX>

Linux history

1990s

- У 1990х роках Лінус Торвальдс написав нове POSIX ядро.
- Багато людей в інтернеті поєднали його з інструментами GNU.
- http://en.wikipedia.org/wiki/Linus_Torvalds
- https://en.wikipedia.org/wiki/History_of_Linux
- <https://en.wikipedia.org/wiki/Linux>
- <https://lwn.net>
- <http://www.levenez.com/unix/> (a huge Unix history poster)

Linux history

Наші дні

- Хто використовує Linux:
 - 97% світових Supercomputers;
 - 80% смартфонів;
 - мільйони PC;
 - 70% веб-серверів;
 - планшетні комп'ютери;
 - побутові прилади;
- <http://kernelnewbies.org/DevelopmentStatistics>
- <http://kernel.org>
- <http://www.top500.org>

Linux history

Дистрибутиви

- Red Hat:
 - Red Hat Enterprise Linux (RHEL);
 - Fedora;
- Ubuntu;
- Debian:
 - Stable;
 - Testing;
 - Unstable;
- І інші:
 - CentOS, Oracle Enterprise Linux, Scientific Linux базуються на RHEL;
 - Linux Mint, Edubuntu, *buntu – базуються на Ubuntu;
 - Сотні інших менш відомих дистрибутивів.

Linux history

Дистрибутиви. Який обрати?

Дистрибутив	Причина використання
Red Hat Enterprise (RHEL)	Для менеджерів. Хороший support від Red Hat за контрактом.
CentOS	Red Hat без support'a.
Fedora	Red Hat для персонального комп'ютера.
Linux Mint	ПК для програвання фільмів, музики, ігор.
Debian	Хороший дистрибутив для серверів, ноутбуків, інших приладів.
Ubuntu	Дуже популярний дистрибутив.
Kali	Дистрибутив для проведення тестів на безпеку, цифрової криміналістики.
Інші	Advanced користувачі віддають перевагу Arch, Gentoo, OpenSUSE, Scientific...

Basics

man

- **man \$command** – для виводу інформації по команді. Для виходу натисніть **q**. Деякі **man** сторінки мають приклади (в кінці).

man route

- **man \$configfile** – деякі конфігураційні файли також мають мануал.

man interfaces

- **man \$daemon** – також є мануал для більшості фонових програмів (daemon)

man syslogd

- **man -k (apropos)** – показує список man сторінок, що містять певний рядок.

man -k syslog

Basics

man

- **whatis** – щоб побачити тільки опис man сторінки.

whatis route

- **whereis** – щоб побачити місцезнаходження man сторінки. Цей файл можна відкрити за допомогою команди man.

whereis -m whatis

- У всіх man сторінок є цифри в дужках. За допомогою команди нижче ви можете дізнатися що це номери секцій.

man man

Basics

Секції man

- Список секцій:
 - 1 Executable programs or shell commands
 - 2 System calls (functions provided by the kernel)
 - 3 Library calls (functions within program libraries)
 - 4 Special files (usually found in /dev)
 - 5 File formats and conventions eg /etc/passwd
 - 6 Games
 - 7 Miscellaneous (including macro packages and conventions), e.g. man(7)
 - 8 System administration commands (usually only for root)
 - 9 Kernel routines [Non standard]

Basics

man

- **man \$section \$file** – тому, коли звертаєтесь до сторінки на команду `passwd` ви побачите `passwd(1)`; коли звертаєтесь до `passwd` файла ви побачите `passwd(5)`.

man passwd – відкриє перший знайдений мануал.

man 5 passwd – відкриє сторінку з секції 5

Basics

mandb

- Якщо ви маєте переконання, що існує сторінка man, але ви не можете отримати доступ до неї, спробуйте запустити **mandb**:

mandb

0 man subdirectories contained newer manual pages.

0 manual pages were added.

0 stray cats were added.

0 old database entries were purged.

Basics

info

- Команда **info** є деякою альтернативою команді **man**. Для отримання інформації по окремій команді треба задати в командному рядку **info** з параметром, що є ім'ям, для команди, що вас цікавить, наприклад:

info man

Basics

Отримання/зміна особистої інформації

- Команда **whoami** повідомляє вам ваше ім'я користувача:

whoami

ruslan

- Команда, **who** дасть вам інформацію про те, хто залогінений в систему:

who

root pts/0 2016-10-10 23:07 (10.104.33.101)

ruslan pts/1 2016-10-10 23:30 (10.104.33.101)

laura pts/2 2016-10-10 23:34 (10.104.33.96)

tania pts/3 2016-10-10 23:39 (10.104.33.91)

Basics

Отримання/зміна особистої інформації

- З **who am i** команда who покаже лише ту лінію, яка вказує на ваш поточний сеанс:

who am i

```
ruslan pts/1 2016-10-10 23:30 (10.104.33.101)
```

- Команда **w** покаже вам, хто залогінений в систему, і що вони роблять:

w

```
23:34:07 up 31 min, 2 users, load average: 0.00, 0.01, 0.02
```

```
USER TTY LOGIN@ IDLE JCPU PCPU WHAT
```

```
root pts/0 23:07 15.00s 0.01s 0.01s top
```

```
ruslan pts/1 23:30 7.00s 0.00s 0.00s w
```

Basics

Отримання/зміна особистої інформації

- Команда **finger** виводить в список для кожного користувача системи UNIX на даний момент ім'я реєстрації в системі, повне ім'я, ім'я терміналу і статус запису (при відсутності дозволу на запис перед термінальним ім'ям вказується символ "*"), час простою, час реєстрації, місце розташування роботи та телефонний номер (якщо вони задані):

finger

```
Login Name Tty Idle Login Time Office Office Phone  
ruslan ruslan *tty1 Jul 30 16:12
```

Basics

Отримання/зміна особистої інформації

- **chfn** змінює повне ім'я користувача, номер робочого місця, додатковий номер робочого місця та номер домашнього телефону в обліковому записі користувача. Ця інформація зазвичай виводиться при запуску програм типу **finger**. Простий користувач може змінювати тільки поля свого облікового запису, суперкористувач може змінювати поля будь-яких записів:

chfn

Basics

Отримання/зміна особистої інформації

- **passwd** змінює (або встановлює) пароль, пов'язаний з ім'ям користувача. Звичайний користувач може змінити тільки пароль, пов'язаний з його власним вхідним_іменем. Суперкористувач (root) має право змінювати будь-які паролі:

passwd

I/O subsystem

FHS

- Більшість дистрибутивів Linux дотримуються FHS (Filesystem Hierarchy Standard).
- FHS доступний за адресою: <http://www.pathname.com/fhs/>

I/O subsystem

man hier

- У файловій системі між дистрибутивами Linux є деякі відмінності. Щоб отримати довідку про вашу машину, введіть **man hier**, щоб знайти інформацію про ієрархію файлової системи. Цей мануал пояснить структуру каталогів на вашому комп'ютері.

I/O subsystem

root директорія

- Всі системи Linux мають структуру каталогів, яка починається з кореневого каталогу. Кореневий каталог представлений прямим слешем: /. Все, що існує у вашій системі Linux, можна знайти нижче цього кореневого каталогу:

ls /

bin dev home media mnt proc sbin srv tftpboot usr
boot etc lib misc opt root selinux sys tmp var

I/O subsystem

Бінарні каталоги

- Двійкові файли (**binaries**) - це файли, що містять зкомпільований вихідний код (або машинний код). Бінарні файли можуть бути виконані на комп'ютері. Іноді **binaries** називаються виконуваними файлами (**executables**).

I/O subsystem

/bin

- У директорії **/bin** містяться **binaries** для використання всіма користувачами.
- Згідно з FHS, каталог `/bin` повинен містити **/bin/cat** і **/bin/date**.

ls /bin

```
archdetect egrep mt setupcon  
autopartition false mt-gnu sh  
bash fgconsole mv sh.distrib  
bunzip2 fgrep nano sleep  
bzcat fuser nc stralign
```

...

I/O subsystem

Інші `/bin` директорії

- Каталог **`/bin`** міститься у багатьох інших каталогах.
- Приклад:
 - Користувач `serena` може встановити свої власні програми в `/home/serena/bin`.

I/O subsystem

/sbin

- **/sbin** містить binaries для налаштування операційної системи.
- Більшість системних файлів вимагають **root** прав.

ls -l /sbin/ifconfig /sbin/fdisk /sbin/mkfs.ext4

-rwxr-xr-x 1 root root 97172 2011-02-02 09:56 /sbin/fdisk

-rwxr-xr-x 1 root root 65708 2010-07-02 09:27 /sbin/ifconfig

-rwxr-xr-x 5 root root 55140 2010-08-18 18:01 /sbin/mkfs.ext4

Конфігураційні каталоги

/boot

- **/boot** містить усі файли, необхідні для завантаження комп'ютера.
- У системах Linux зазвичай ви знайдете каталог `/boot/grub`.

Конфігураційні каталоги

/etc

- **/boot** містить усі файли специфічних налаштувань комп'ютера.
- Історично /etc це etcetera (тобто, тощо) але зараз часто використовують Editable Text Configuration.
- Імена файлів конфігурації часто співпадають з ім'ям додатку, протокола і т.д. з розширенням .conf .

ls /etc/*.conf

```
/etc/adduser.conf /etc/ld.so.conf /etc/scrollkeeper.conf  
/etc/brltty.conf /etc/lftp.conf /etc/sysctl.conf  
/etc/ccertificates.conf /etc/libao.conf /etc/syslog.conf  
/etc/cvs-cron.conf /etc/logrotate.conf /etc/ucf.conf  
/etc/ddclient.conf /etc/ltrace.conf /etc/uniconf.conf
```

...

Каталоги даних

/home

- **/home** каталог для зберігання особистих даних.
- Поширена практика називати домашній каталог у форматі `/home/$USERNAME`.
- Приклад:

ls /home

```
geert annik sandra paul tom
```

- Також **/home** служить для зберігання профілю користувача.
- Профіль користувача зазвичай містить багато прихованих файлів, в яких містяться налаштування користувача.

Каталоги даних

/home

ls -d /home/ruslan/*

```
/home/ruslan/. /home/ruslan/.bash_profile /home/ruslan/.ssh  
/home/ruslan/.. /home/ruslan/.bashrc /home/ruslan/.viminfo  
/home/ruslan/.bash_history /home/ruslan/.lesshst
```

Каталоги даних

/root

- **/root** - це місце розташування за замовчуванням для персональних даних та профілю root користувача.

Каталоги даних

/media

- **/media** служить точкою монтування для знімних носіїв.
- Приклади:
 - CD-rom;
 - Цифрові камери;
 - Різні USB-пристрої;

Каталоги даних

`/mnt`

- **`/mnt`** має бути порожнім та використовуватись лише для тимчасових точок монтування (відповідно до FHS).

Каталоги даних

/tmp

- **/tmp** для зберігання тимчасових даних, коли це необхідно.
- Ніколи не використовуйте **/tmp** для зберігання даних, які важливі або які ви хочете зберегти.



I/O subsystem

...та інші

I/O subsystem

pwd

- **pwd** – команда для виводу вашого поточного каталогу.

pwd

/home/ruslan

I/O subsystem

pwd

- **pwd** – команда для виводу вашого поточного каталогу.

pwd

/home/ruslan

I/O subsystem

cd

- **cd** – команда для зміни поточного каталогу.

cd /etc

pwd

/etc

cd /bin

pwd

/bin

cd /home/ruslan/

pwd

/home/ruslan

I/O subsystem

`cd ~`

- За допомогою **cd** можна повернутися до вашого домашнього каталогу. **cd ~** має той самий ефект.

cd /etc

pwd

/etc

cd

pwd

/home/ruslan

cd ~

pwd

/home/ruslan

I/O subsystem

cd -

- Для повернення до попереднього каталогу введіть **cd -**

pwd

/home/ruslan

cd /etc

pwd

/etc

cd -

/home/ruslan

cd -

/etc

I/O subsystem

Абсолютні та відносні шляхи

pwd

/home/ruslan

cd home

bash: cd: home: No such file or directory

cd /home

pwd

/home

pwd

/home

cd /ruslan

bash: cd: /ruslan: No such file or directory

cd ruslan

pwd

/home/ruslan

I/O subsystem

ls

- **ls** – команда для відображення вмісту каталогу.

ls

allfiles.txt dmesg.txt services stuff summer.txt

- **ls -a** – команда для відображення всіх файлів, включаючи приховані.

ls -a

allfiles.txt .bash_profile dmesg.txt .lesshst stuff
.bash_history .bashrc services .ssh summer.txt

- **ls -l** – команда

I/O subsystem

ls -l

- **ls -l** – команда для відображення файлів та їх параметрів.

ls -l

total 17296

-rw-r--r-- 1 ruslan ruslan 17584442 Sep 17 00:03 allfiles.txt

-rw-r--r-- 1 ruslan ruslan 96650 Sep 17 00:03 dmesg.txt

-rw-r--r-- 1 ruslan ruslan 19558 Sep 17 00:04 services

drwxr-xr-x 2 ruslan ruslan 4096 Sep 17 00:04 stuff

-rw-r--r-- 1 ruslan ruslan 0 Sep 17 00:04 summer.txt

I/O subsystem

ls -lh

- **ls -l** – команда для відображення файлів та їх параметрів.

ls -l -h

total 17M

-rw-r--r-- 1 ruslan ruslan 17M Sep 17 00:03 allfiles.txt

-rw-r--r-- 1 ruslan ruslan 95K Sep 17 00:03 dmesg.txt

-rw-r--r-- 1 ruslan ruslan 20K Sep 17 00:04 services

drwxr-xr-x 2 ruslan ruslan 4.0K Sep 17 00:04 stuff

- Також можливі такі варіанти:

ls -lh

ls -hl

ls -h -l

I/O subsystem

mkdir

- **mkdir** – команда для створення власних каталогів.

```
mkdir mydir
```

```
cd mydir
```

```
ls -al
```

```
total 8
```

```
drwxr-xr-x 2 ruslan ruslan 4096 Sep 17 00:07 .
```

```
drwxr-xr-x 48 ruslan ruslan 4096 Sep 17 00:07 ..
```

```
mkdir stuff
```

```
mkdir otherstuff
```

```
ls -l
```

```
total 8
```

```
drwxr-xr-x 2 ruslan ruslan 4096 Sep 17 00:08 otherstuff
```

```
drwxr-xr-x 2 ruslan ruslan 4096 Sep 17 00:08 stuff
```

I/O subsystem

mkdir -p

- Не вдається виконати наступну команду, оскільки батьківський каталог **threedsdeep** не існує:

```
mkdir mydir2/mysubdir2/threedsdeep
```

```
mkdir: cannot create directory 'mydir2/mysubdir2/threedsdeep': No  
such file or directory
```

- Якщо вказати опцію -p, то mkdir буде створювати батьківські каталоги за потреби:

```
mkdir -p mydir2/mysubdir2/threedsdeep
```

I/O subsystem

rmmdir

- **rmmdir** – команда для видалення каталогів.

rmmdir otherstuff

cd ..

rmmdir mydir

rmmdir: failed to remove 'mydir': Directory not empty

- І, подібно до опції **mkdir -p**, ви також можете використовувати **rmmdir** для рекурсивного видалення каталогів.

I/O subsystem

touch

- **touch** – команда для створення файлів.

ls -l

total 0

touch file42

touch file33

ls -l

total 0

-rw-r--r-- 1 ruslan ruslan 0 Oct 15 08:57 file33

-rw-r--r-- 1 ruslan ruslan 0 Oct 15 08:56 file42

I/O subsystem

touch -t

- **touch -t** – команда для створення файлів з вказуванням дати створення.

touch -t 201805050000 SinkoDeMayo

ls -l

-rw-r--r-- 1 ruslan ruslan 0 May 5 2018 SinkoDeMayo

I/O subsystem

rm

- **rm** – команда для видалення файлів (не в корзину).
- **rm -i** – команда для видалення файлів з запитом на видалення.
- **rm -rf** – команда для видалення будь-якого каталогу (навіть не порожнього).

I/O subsystem

cp

- **cp** – команда для копіювання:

cp file42 file42.copy

- Якщо ціль є каталогом, то файл скопіюється до нього:

cp file42 dir42

- **cp -r** – команда для копіювання каталогів (рекурсивне копіювання всіх файлів у всіх підкаталогах):

cp -r dir42/ dir33

- Можна копіювати відразу багато файлів у каталог:

cp file42 file42.copy SinkoDeMayo dir42/

- Щоб запобігти перезаписуванню існуючих файлів, скористайтеся опцією **-i**:

cp -i SinkoDeMayo file42

I/O subsystem

mv

- **mv** – команда для перейменування або переміщення файлу в інший каталог:

mv file42 file33

- Також можна перейменовувати каталоги:

mv dir33 backup

- **mv -i** – команда, подібна як в `cp` і `rm` для запобігання випадкового перезаписування існуючого файлу:

mv -i file33 SinkoDeMayo

mv: overwrite `SinkoDeMayo'? **no**

I/O subsystem

cat

- **cat** – це один з найбільш універсальних інструментів. Все що він робить це копіює стандартний **input** до стандартного **output**.
- Можна використовувати для відображення файлу на екран:

cat /etc/resolv.conf

- Можна об'єднувати вивід з файлів:

cat part1 part2 part3

cat part1 part2 part3 >all

cat all

I/O subsystem

cat

- За допомогою **cat** можна створювати текстові файли і вносити відразу туди текст:

cat > winter.txt

- Просто натискайте Enter по закінченню кожного рядка. Для закінчення редагування файла натисніть Ctrl d.

I/O subsystem

cat

- Ви можете вибрати маркер закінчення для **cat** за допомогою << :

cat > hot.txt <<stop

> It is hot today!

> Yes it is summer.

> stop

cat hot.txt

It is hot today!

Yes it is summer.

I/O subsystem

cat

- За допомогою **cat** ви можете копіювати файли:

cat winter.txt

It is very cold today!

cat winter.txt > cold.txt

cat cold.txt

It is very cold today!

I/O subsystem

tac

- Також є зворотній **cat** - **tac**:

cat count

one

two

three

four

tac count

four

three

two

one

I/O subsystem

more та less

- **more** – команда для відображення файлу, що займає більше одного екрану.
- **less** – схожа на **more**, тільки працює за іншим механізмом (покращена версія more з можливістю зворотної прокрутки).

I/O subsystem

grep

- **grep** – фільтр відомий серед користувачів Unix. Найпоширенішим використанням grep є фільтрація рядків тексту, що містить (або не містить) певного рядка:

cat tennis.txt

Amelie Mauresmo, Fra

Kim Clijsters, BEL

Justine Henin, Bel

Serena Williams, usa

Venus Williams, USA

cat tennis.txt | grep Williams

Serena Williams, usa

Venus Williams, USA

I/O subsystem

grep

- Можливо писати без **cat**:

grep Williams tennis.txt

Serena Williams, usa

Venus Williams, USA

I/O subsystem

grep

- **grep -i** – фільтрує регістронезалежним способом.
- **grep -v** – виводить рядки, які не відповідають вибраному слову.
- **grep -A1** – до шуканого рядка ще додається вивід наступного рядка.
- **grep -B1** – до шуканого рядка ще додається вивід попереднього рядка.
- **grep -C1** – до шуканого рядка ще додається вивід попереднього і наступного рядка.

I/O subsystem

find

- **find** – команда для пошуку файлів.
- Приклади:
 - Знайти всі файли в /etc і покласти список у файл etcfiles.txt:
find /etc > etcfiles.txt
 - Пошук по імені файлу:
find . -name "*.conf"
 - Ще багато різних.

- Список всіх опцій є тут:

<https://rtfm.co.ua/komanda-find-i-eyo-opcii-v-primerax/>

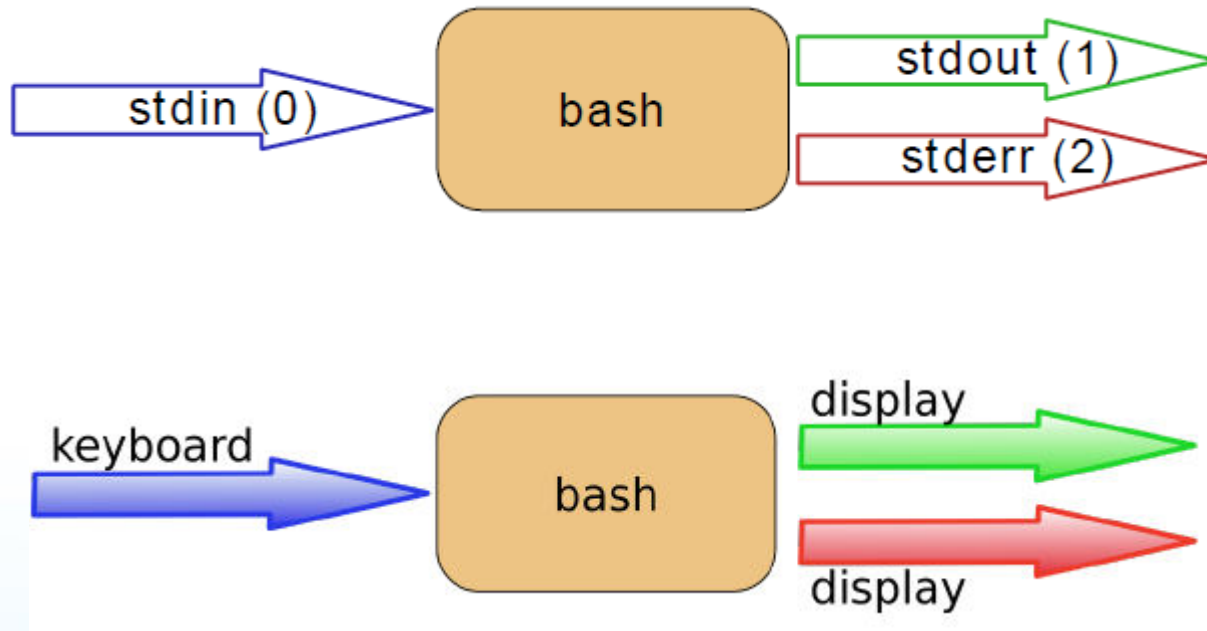
I/O subsystem

locate

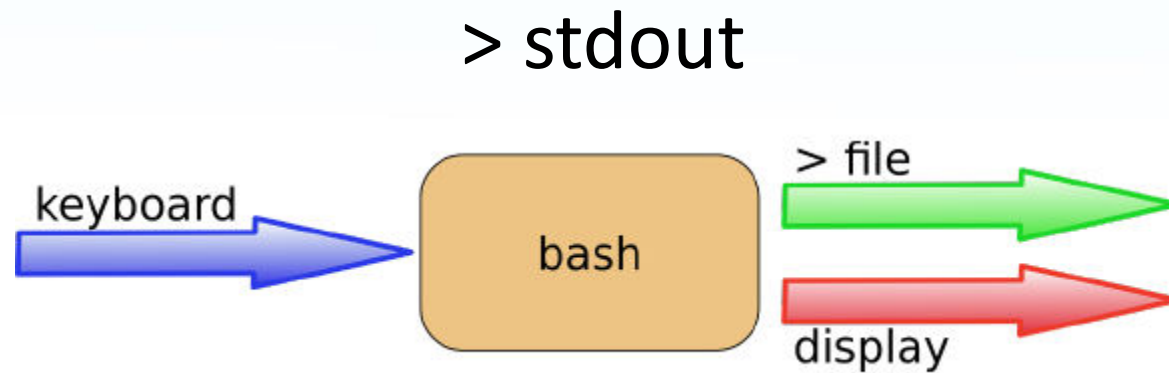
- **locate** – також команда для пошуку файлів. Використовує власну базу даних для пошуку, що є швидше ніж команда **find**.
- Для отримання результатів, які відповідають дійсності, треба тримати в оновленому стані базу даних зі списком імен файлів.
- Для оновлення періодично виконуйте команду **updatedb**.

I/O subsystem

stdin, stdout, stderr



I/O subsystem



- > є абрєвіатурою 1> (stdout називається потоком 1)

echo It is cold today! > winter.txt

cat winter.txt

It is cold today!

I/O subsystem

> stdout

- Під час сканування рядка, оболонка побачить знак > і очистить файл! Оскільки це відбувається, перш ніж виконується аргумент 0, це означає, що навіть коли команда завершиться помилкою, файл буде очищено!

cat winter.txt

It is cold today!

zcho It is cold today! > winter.txt

-bash: zcho: command not found

cat winter.txt

I/O subsystem

noclobber

- Видалення вмісту файлу при використанні оператора > може бути припинено шляхом установки параметра командної оболонки **noclobber**:

cat winter.txt

It is cold today!

set -o noclobber

echo It is cold today! > winter.txt

-bash: winter.txt: cannot overwrite existing file

set +o noclobber

I/O subsystem

Нейтралізація noclobber

- Вплив параметра командної оболонки noclobber може бути нейтралізовано за допомогою оператора >|:

```
set -o noclobber
```

```
echo It is cold today! > winter.txt
```

```
-bash: winter.txt: cannot overwrite existing file
```

```
echo It is very cold today! >| winter.txt
```

```
cat winter.txt
```

```
It is very cold today!
```

I/O subsystem

>> append

- Для додавання виводу у файл використовуйте >>:

```
echo It is cold today! > winter.txt
```

```
cat winter.txt
```

It is cold today!

```
echo Where is the summer ? >> winter.txt
```

```
cat winter.txt
```

It is cold today!

Where is the summer ?

I/O subsystem

2> stderr



- Переадресація **stderr** виконується за допомогою 2>

find / > allfiles.txt 2> /dev/null

I/O subsystem

2>&1

- Щоб перенаправити **stdout** і **stderr** до одного файлу, використовуйте **2>&1**:

find / > allfiles_and_errors.txt 2>&1

I/O subsystem

< stdin

- Перенаправлення **stdin** виконується за допомогою < (скорочений запис 0<).:

```
cat < text.txt
```

```
one
```

```
two
```

```
tr 'onetw' 'ONEZZ' < text.txt
```

```
ONE
```

```
ZZO
```


I/O subsystem

<< here document

- Спосіб вводу до тих пір, поки з'явиться певна послідовність символів (зазвичай EOF або явно через Ctrl d):

```
cat <<EOF > text.txt
```

```
> one
```

```
> two
```

```
> EOF
```

```
cat text.txt
```

```
one
```

```
two
```

```
cat <<brol > text.txt
```

```
> brel
```

```
> brol
```

```
cat text.txt
```

```
brel
```

I/O subsystem

<<< here string

- <<< можна використовувати для прямого передавання рядків до команди. Результат аналогічний тому, коли використовуєте **echo string | command** (але так на один запущений процес менше):

```
base64 <<< linux-training.be
```

```
bGludXgtdHJhaW5pbmcuYmUK
```

```
base64 -d <<< bGludXgtdHJhaW5pbmcuYmUK
```

```
linux-training.be
```