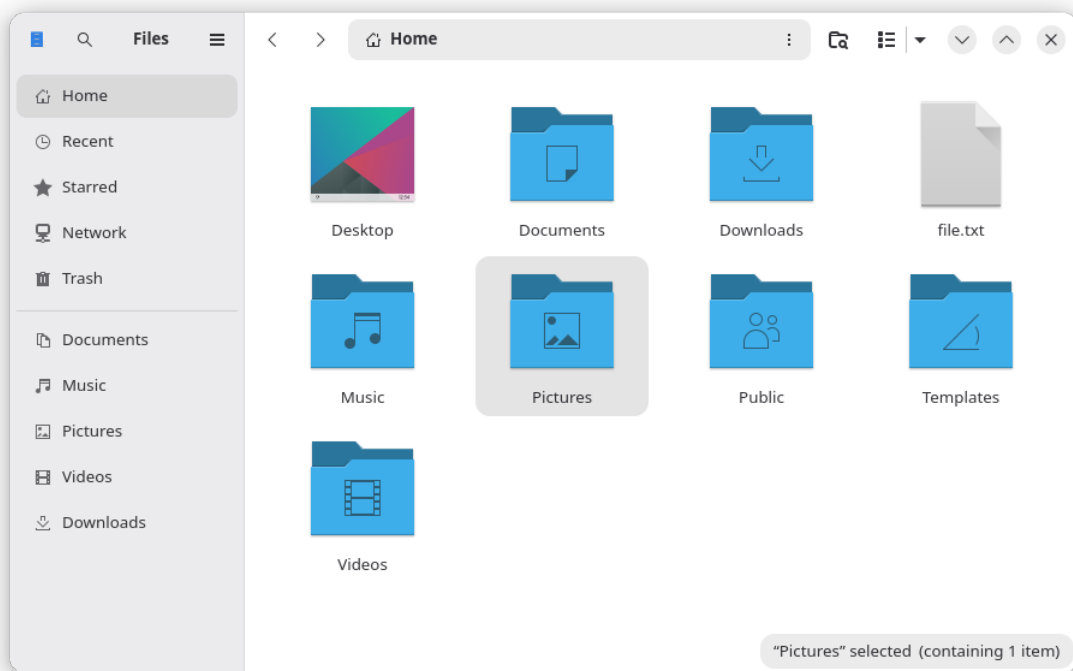


Współczesne systemy komputerowe

Praca z plikami i katalogami

Interfejs graficzny

- W powłoce graficznej GNOME do pracy z plikami i katalogami można użyć dedykowanego domyślnego narzędzia **Files**. Jego działanie jest analogiczne do Eksploratora plików w systemach z rodziny Windows.



Katalogi

- Zmiana katalogu, polecenie **cd**.

cd <i>name</i>	zmiana katalogu na <i>name</i>
cd /etc	zmiana katalogu na /etc (ścieżka bezwzględna)
cd	zmiana katalogu na domowy
cd ..	przejdź do katalogu wyżej
cd ../..	przejdź dwa katalogi wyżej
cd -	przejdź do ostatniego katalogu

```
foo@debian:~$ cd /var
```

```
foo@debian:/var$ pwd
/var
```

```
foo@debian:/var$ cd
```

```
foo@debian:~$ pwd
```

```
| /home/foo
```

- Listowanie zawartości katalogu, polecenie **ls**.

brak	wyświetla zawartość katalogu w kilku kolumnach
-a	wyświetla pliki ukryte (z kropką na początku nazwy)
-F	dodatkowo oznacza typ pliku (/ - katalog, * - program lub skrypt wykonywalny, - plik FIFO, @ - link symboliczny)
-l	lista szczegółowa (prawa dostępu, czas modyfikacji, rozmiar)
-t	lista posortowana według daty zmiany
-r	odwraca kolejność wyświetlania listy
-R	uwzględnia podkatalogi
-u	lista posortowana według daty dostępu

```
foo@debian:~$ ls /var
backups  cache  lib  local  lock  log  mail  opt  run  spool  tmp
```

```
foo@debian:~$ ls -l /var
total 36
drwxr-xr-x  2 root root 4096 Nov 28 01:38 backups
drwxr-xr-x 21 root root 4096 Nov 27 08:50 cache
drwxr-xr-x 55 root root 4096 Nov 27 05:21 lib
drwxr-xr-x  2 root root 4096 Nov  7 12:40 local
lrwxrwxrwx  1 root root    9 Nov 27 04:56 lock -> /run/lock
drwxr-xr-x 10 root root 4096 Nov 28 07:15 log
drwxrwsr-x  2 root mail 4096 Nov 27 04:56 mail
drwxr-xr-x  2 root root 4096 Nov 27 04:56 opt
lrwxrwxrwx  1 root root    4 Nov 27 04:56 run -> /run
drwxr-xr-x  6 root root 4096 Nov 27 05:09 spool
drwxrwxrwt 12 root root 4096 Nov 28 07:16 tmp
```

- Polecenie **pwd** wyświetla aktualny katalog.

```
foo@debian:~$ cd /
```

```
foo@debian:/$ pwd
/
```

```
foo@debian:/$ ls
bin  etc          initrd.img.old  lost+found  opt  run  sys  var
boot home        lib             media       proc sbin tmp  vmlinuz
dev  initrd.img  lib64          mnt         root srv  usr  vmlinuz.old
```

```
foo@debian:/$ cd /var/spool
```

```
foo@debian:/var/spool$ pwd
/var/spool
```

```
foo@debian:/var/spool$ ls
anacron  cron  cups  libreoffice  mail
```

```
foo@debian:/var/spool$ cd
```

- Katalogi można tworzyć poleceniem **mkdir**, polecenie **mkdir -p** utworzy konieczne podkatalogi.

```
foo@debian:~$ mkdir dir2
```

```
foo@debian:~$ mkdir dir{3,4,5}

foo@debian:~$ mkdir -p dir1/subdir{1,2,3}

foo@debian:~$ ls -l
...
drwxrwxr-x 5 foo foo 4096 Nov 28 07:25 dir1
drwxrwxr-x 2 foo foo 4096 Nov 28 07:24 dir2
drwxrwxr-x 2 foo foo 4096 Nov 28 07:25 dir3
drwxrwxr-x 2 foo foo 4096 Nov 28 07:25 dir4
drwxrwxr-x 2 foo foo 4096 Nov 28 07:25 dir5
...
```

```
foo@debian:~$ ls dir1
subdir1 subdir2 subdir3
```

```
foo@debian:~$ ls dir1
subdir1 subdir2 subdir3
```

- Kasowanie pustych katalogów, polecenie **rmdir**.

```
foo@debian:~$ ls dir1 dir2
dir1:
subdir1 subdir2 subdir3

dir2:
```

```
foo@debian:~$ cd dir1
```

```
foo@debian:~/dir1$ ls -l
total 12
drwxrwxr-x 2 foo foo 4096 Nov 28 07:25 subdir1
drwxrwxr-x 2 foo foo 4096 Nov 28 07:25 subdir2
drwxrwxr-x 2 foo foo 4096 Nov 28 07:25 subdir3
```

```
foo@debian:~/dir1$ rmdir subdir1
```

```
foo@debian:~/dir1$ rmdir subdir2 subdir3
```

```
foo@debian:~/dir1$ ls
```

```
foo@debian:~/dir1$ cd ..
```

```
foo@debian:~$ rmdir dir{2,3,4,5}
```

Pliki

- Polecenie **touch** służy do zmiany czasu ostatniego otwarcia pliku lub do tworzenia pliku o wielkości 0 B.

-a	zmiana czasu ostatniego otwarcia pliku
-m	zmiana czasu ostatniej modyfikacji pliku
-r <i>file</i>	używa czasów pliku <i>file</i> zamiast aktualnego
-t <i>time</i>	używa czasu <i>time</i> , zamiast czasu aktualnego

([[[CC]YY]MMDDhhmm.[ss] ([Century]Year) Month Day Hour Minute [Seconds]])

```
foo@debian:~$ ls -lat
total 112
drwx----- 21 foo  foo  4096 Nov 28 07:25 .
drwxrwxr-x  5 foo  foo  4096 Nov 28 07:25 dir1
drwxrwxr-x  2 foo  foo  4096 Nov 28 07:25 dir3
...
-rw-r--r--  1 foo  foo   807 Nov 27 05:14 .profile
drwxr-xr-x  3 root root 4096 Nov 27 05:14 ..
```

```
foo@debian:~$ touch .profile
```

```
foo@debian:~$ ls -lat
total 112
-rw-r--r--  1 foo  foo   807 Nov 28 07:41 .profile
drwx----- 21 foo  foo  4096 Nov 28 07:25 .
drwxrwxr-x  5 foo  foo  4096 Nov 28 07:25 dir1
...
drwxr-xr-x  3 root root 4096 Nov 27 05:14 ..
```

```
foo@debian:~$ touch new-file
```

```
foo@debian:~$ ls -lat
total 112
-rw-rw-r--  1 foo  foo    0 Nov 28 07:43 new-file
drwx----- 21 foo  foo  4096 Nov 28 07:43 .
-rw-r--r--  1 foo  foo   807 Nov 28 07:41 .profile
...
drwxr-xr-x  3 root root 4096 Nov 27 05:14 ..
```

- Do wyświetlania zawartości plików można użyć polecenia **cat**.

```
foo@debian:~$ cat /etc/hostname
debian
```

- Do wyświetlania zawartości plików można też używać polecenia **less**, które jest również używane do wyświetlania manuali (patrz wskazówki powyżej).
- Do wyświetlania kilku pierwszych lub ostatnich linii pliku można stosować polecenia **head** i **tail**, które domyślnie wyświetlają 10 pierwszych lub ostatnich linii pliku, liczbę linii można podać jako parametr.

```
foo@debian:~$ head /etc/debconf.conf
# This is the main config file for debconf. It tells debconf where to
# store data. The format of this file is a set of stanzas. Each stanza
# except the first sets up a database for debconf to use. For details, see
# debconf.conf(5) (in the debconf-doc package).
#
# So first things first. This first stanza gives the names of two databases.

# Debconf will use this database to store the data you enter into it,
# and some other dynamic data.
Config: configdb
```

```
foo@debian:~$ tail .bash_history
ls -lat
cd ..
ls -lat
touch .profile
ls -lat
touch new-file
ls -lat
cat /etc/hostname
head /etc/debconf.conf
tail .bash_history
```

- Polecenie **tail -f file** pozwala na ciągły podgląd pliku file (zwykle używane przy obserwowaniu logów), działanie programu można zakończyć kombinacją klawiszy [CTRL+C].
- Do przenoszenia plików (i zmiany nazwy) można użyć polecenia **mv**.

```
foo@debian:~$ touch dir1/file{1,2,3}.txt
```

```
foo@debian:~$ ls dir1
file1.txt file2.txt file3.txt
```

```
foo@debian:~$ cd dir1
```

```
foo@debian:~/dir1$ ls
file1.txt file2.txt file3.txt
```

```
foo@debian:~/dir1$ mkdir subdir1
```

```
foo@debian:~/dir1$ mv *.txt subdir1/
```

```
foo@debian:~/dir1$ ls
subdir1
```

```
foo@debian:~/dir1$ cd subdir1/
```

```
foo@debian:~/dir1/subdir1$ ls
file1.txt file2.txt file3.txt
```

```
foo@debian:~/dir1/subdir1$ mv file1.txt file-a.txt
```

```
foo@debian:~/dir1/subdir1$ mv file2.txt file-b.txt
```

```
foo@debian:~/dir1/subdir1$ mv file3.txt file-c.txt
```

```
foo@debian:~/dir1/subdir1$ ls
file-a.txt file-b.txt file-c.txt
```

- Do kopiowania plików służy polecenie **cp**.

-a	kopiowanie katalogów z podkatalogami, linków symbolicznych, prawa dostępu, właściciele plików i czasy nie są zmieniane
-i	pytanie przed nadpisaniem pliku
-R, -r	kopiowanie rekursywne katalogów (katalogi z podkatalogami)
-s	tworzenie linków symbolicznych zamiast kopiowania
-l	tworzenie linków zamiast kopiowania
-u	kopiowanie plików nowszych niż docelowe lub gdy docelowe nie istnieją

```
foo@debian:~/dir1/subdir1$ ls
file-a.txt file-b.txt file-c.txt

foo@debian:~/dir1/subdir1$ cd ..

foo@debian:~/dir1$ ls
subdir1

foo@debian:~/dir1$ mkdir subdir2

foo@debian:~/dir1$ ls
subdir1 subdir2

foo@debian:~/dir1$ cp -r subdir1/* subdir2/

foo@debian:~/dir1$ ls subdir2/
file-a.txt file-b.txt file-c.txt
```

- Kasowanie plików i katalogów, polecenie **rm**.

-i	pytanie o potwierdzenie skasowania pliku
-r	kasowanie rekursywne (katalogi z podkatalogami)
-f	kasowanie bez pytania plików tylko do odczytu

```
foo@debian:~/dir1$ cd ..

foo@debian:~$ ls dir1
subdir1 subdir2

foo@debian:~$ ls dir1/subdir1
file-a.txt file-b.txt file-c.txt

foo@debian:~$ rm dir1/subdir1/file-a.txt

foo@debian:~$ rm dir1/subdir1/file-b.txt

foo@debian:~$ rm dir1/subdir1/file-c.txt

foo@debian:~$ ls dir1/subdir2
file-a.txt file-b.txt file-c.txt

foo@debian:~$ rm -r dir1/
```

Linki

- Linki, polecenie **ln**.

```
foo@debian:~$ mkdir links

foo@debian:~$ cd links

foo@debian:~/links$ touch file

foo@debian:~/links$ ls -li
total 0
11873 -rw-r--r-- 1 foo foo 0 Feb 28 17:16 file

foo@debian:~/links$ ln file hard-link

foo@debian:~/links$ ls -li
```

```
total 0
11873 -rw-r--r-- 2 foo foo 0 Feb 28 17:16 file
11873 -rw-r--r-- 2 foo foo 0 Feb 28 17:16 hard-link
```

```
foo@debian:~/links$ ln -s file symbolic-link
```

```
foo@debian:~/links$ ls -li
total 0
11873 -rw-r--r-- 2 foo foo 0 Feb 28 17:16 file
11873 -rw-r--r-- 2 foo foo 0 Feb 28 17:16 hard-link
13639 lrwxrwxrwx 1 foo foo 4 Feb 28 17:17 symbolic-link -> file
```

- Linki można kasować poleceniem **rm**, nie można rozróżnić pliku i linku twardego.

```
foo@debian:~/links$ ls -li
total 0
11873 -rw-r--r-- 2 foo foo 0 Feb 28 17:16 file
11873 -rw-r--r-- 2 foo foo 0 Feb 28 17:16 hard-link
13639 lrwxrwxrwx 1 foo foo 4 Feb 28 17:17 symbolic-link -> file
```

```
foo@debian:~/links$ rm symbolic-link
```

```
foo@debian:~/links$ ls -li
total 0
11873 -rw-r--r-- 2 foo foo 0 Feb 28 17:16 file
11873 -rw-r--r-- 2 foo foo 0 Feb 28 17:16 hard-link
```

```
foo@debian:~/links$ rm file
```

```
foo@debian:~/links$ ls -li
total 0
11873 -rw-r--r-- 1 foo foo 0 Feb 28 17:16 hard-link
```