Ten program już się kilka razy pojawił, ale warto poświęcić mu więcej czasu bo ma ogromne możliwości. Oczywiście zachęcam do lektury manuala w systemie oraz opisów znalezionych w sieci. Pomiędzy różnymi systemami mogą być drobne odstępstwa w zachowaniu find, ale dla nas raczej są one do pominięcia. Skupimy się na podstawach, które są uniwersalne.
Generalnie program `find` służy do szukania plików w systemie. Kryteria mogą być przeróżne jak sami zobaczycie. Warto tutaj od razu zaznaczyć, że przy pomocy `find` można uzyskać dostęp do zawartości pliku wykonując dodatkowe polecenie, ale `find` jako taki widzi tylko nazwy plików, nie ich zawartość. Do szukania plików po zawartości służą inne programy: `fgrep`, `grep` , `egrep` , `awk`.
Program `find` ma zawsze jeden obowiązkowy argument - katalog, od którego ma zacząć poszukiwania. To jest zawsze pierwszy parametr:

```
find /etc
```

Wypisze ścieżki wszystkich plików (podkatalogów, linków, soketów, itd.) poniżej katalogu `/etc`.

Program find ma wiele opcji. Przeanalizujemy tylko te najważniejsze:

`-type` : typ pliku: f - zwykły plik, d - katalog, l - link symboliczny itd.

```
find /etc/ -type d
```

Wypisze tylko podkatalogi poniżej `/etc`.

`-name` : nazwa pliku, można używać *, ?, [,], ale trzeba te znaki chronić używając cydzysłowów

```
find /etc -name "*.conf"
```

Odnajdzie sieżki plików poniżej `/etc`, które na końcu nazwy mają `.conf` (można o tym mysleć jak o rozszerzeniu dosowym, ale w Unixie nie ma takiego pojęcia - to jest część nazwy pliku, a kropek w nazwie może być wiele).

`-mtime` : czas modyfikacji pliku, N - dokładnie N dni temu, -N mniej niż N dni temu, +N więcej niż N dni temu

```
find /etc -mtime -5
```

Wypisze pliki zmodyfikowane 4 lub mniej dni temu.

`-ctime` : j/w tylko czas utworzenia pliku

`-atime` : j/w tylko czas dostępu do pliku

`-newer` : pliki nowsze od wskazanego pliku

```
find /etc/ -newer /tmp/cosik
```

Wypisze pliki poniżej `/etc`, nowsze od pliku `/tmp/cosik`.

`-user` : określa właściciela pliku

```
find /etc/ -user root
```

Wyświetli pliki poniżej /etc, których właścicielem jest użytkownik `root`

`-group` : określa grupę własciciela pliku

```
find /etc/ -group bin
```

Wyświetli pliki poniżej `/etc`, któryw należą do grupy `bin`.

`-perm` : określa prawa dostępu do pliku, temu poświęcimy więcej czasu później

`-ls` : wypisuje pełne dane dla każdego pliku spełniającego kryteria

```
find /etc -ls
```

Wypisze wszystkie pliki poniżej /etc z pełną informacją jak `ls -l`

`-exec` : dla każdego pliku spełniającego kryteria wykonuje dowolny program, w miejsce nawiasów klamrowych `{}` zostanie podstawiona ścieżka pliku, koniec poleceniea tzreba zakończyć `\;`
Opcje -ls i -exec nieco róznią się od wymienionych wcześniej. Pozostałe opcje dla każdego pliku dają prawda (1) lub fałsz (0) w zależnopści czy kryterium jest spełnione. Opcje -ls i -exec zawsze dają prawda. W ten sposób możemy rózne opcje find łączyć ze sobą by budować nawet bardzo skomplikowane wyrażenia logiczne.

```
find /etc -type d -name "in.*"
```

Wyświetli tylko katalog, które mają na początku i`n`. Brak spójnika pomiędzy opcjami jest traktowany jak koniunkcja. Operatorem koniunkcji jest `-a`:

```
find /etc -type d -a -name "in.*"
```

To jest to samo co wyżej. Operator `-o` to alternatywa

```
find /etc -type d -o -name "in*"
```

Wyświetli katalogi (wszystkie) i te pliki które mają `in` na początku.
Znak ! umieszczony przed wyrażeniem z opcją powoduje zanegowanie tego wyrażenia.

```
find /etc ! -type f -name "in*"
```

Wyświetli pliki nie będące plikami zwykłymi o nazwie zaczynającej się na `in`.
Do budowy bardziej skomplikowanych wrażeń logicznych możemy używać nawiasów `()`, ale ponieważ mają one specjalne znaczenie w shellu muszą być ochronione jak średnik w `-exec`.

```
find /etc  \( -type d -name "in*" \) -o \(-type f -name "*.conf" \)
```

Wyświetli katalogi, których nazwy zaczynają się na `in` oraz pliki zwykłe z `.conf` w nazwie na końcu.

Uwaga: Jeśli eksperymentujecie jako zwykły użytkownik to `find` może zgłaszać *Permission denied* bo nie do wszystkich plików macie wtedy dostęp. Na przykład :

```
find /etc -name "inet*"
/etc/default/inetinit
/etc/inet
find: cannot read dir /etc/inet/secret: Permission denied
/etc/inet/inetd.conf
find: cannot read dir /etc/sfw/openssl/private: Permission denied
find: cannot read dir /etc/apache/ssl.key: Permission denied
find: cannot read dir /etc/flash/precreation: Permission denied
find: cannot read dir /etc/webmin: Permission denied
```

```
/etc/inetd.conf
```

Można tego się łatwo pozbyć używając przekierowań bo mamy tu przykład , gdzie standarowe wyjście miesza się z błędami. Należy deskryptor 2 (błędy) przekierować do czarnej dziury. Czarną dziurą w Unixie jest urządzenie `/dev/null`.

```
find /etc -name "inet*" 2> /dev/null
/etc/default/inetinit
/etc/inet
/etc/inet/inetd.conf
/etc/inetd.conf
```

Dla porównania, gdy wyrzucimy deskryptor 1 (wyjście) to zostaną tylko błędy z deskryptora 2:

```
find /etc -name "inet*" 1> /dev/null
find: cannot read dir /etc/inet/secret: Permission denied
find: cannot read dir /etc/sfw/openssl/private: Permission denied
find: cannot read dir /etc/apache/ssl.key: Permission denied
find: cannot read dir /etc/flash/precreation: Permission denied
find: cannot read dir /etc/webmin: Permission denied
```