



# Podstawy systemu Linux

## 05. Analiza logów





# Podstawy systemu Linux

## 05. Analiza logów

### Spis treści

Diagnostyka systemu	4
Zawartość dzienników	4
Przeglądanie dzienników	5
Wybieranie niektórych wpisów	6
Filtrowanie logów z użyciem wyrażeń	8
Rozszerzone wyrażenia regularne	9
Zaawansowane opcje grep	12
Statystyki plików dziennika	14
Spis komend	15

## Diagnostyka systemu

W systemie uruchomione jest wiele programów, które działają, ale ich „nie widać”. Takimi programami może być jądro systemu, program obsługujący system plików, sprawdzający stan połączenia z internetem, program do zarządzania zalogowanymi użytkownikami, albo środowisko graficzne.

Jeżeli pojawi się błąd, albo coś działa nieprawidłowo, ciężko jest natychmiast stwierdzić, co jest przyczyną. Często z góry się zakłada, że po prostu „komputer nie działa”, albo „system ma jakiś problem”. Zazwyczaj wtedy przyczyną może być nie „system”, a na przykład stary dysk twardy, w którym uszkodzeniu uległy części mechaniczne, albo na przykład pojedynczy program, który nie potrafi poradzić sobie z konfiguracją.

Aby zbadać przyczynę problemu, należy przeprowadzić diagnostykę systemu. Tylko jak stwierdzić na pewno, że to problem dysku twardego albo enigmatycznie nazwanego procesu „rtkit-daemon”?

Dużym ułatwieniem diagnostyki byłoby narzędzie, które dostarczy informacji o zdarzeniach w systemie. Na podstawie analizy zdarzeń można stwierdzić, kiedy wystąpił problem i w którym miejscu (np. podczas pracy procesu „rtkit-daemon”).

Takim narzędziem są **dzienniki**, nazywane też logami. Dzienniki dostarczają informacji o zdarzeniach w uruchomionych programach, w tym także o błędach. Zazwyczaj te wpisy zawierają też dodatkowe informacje, które ułatwiają diagnostykę.

## Zawartość dzienników

Dzienniki są użyteczne, ponieważ nie tylko zawierają informacje o błędach, ale także o wszystkich zdarzeniach, które dotyczą danego programu.

Jest to o tyle użyteczne, że dzięki temu można również stwierdzić, że program działa, albo że program wykonał coś, czego się spodziewaliśmy. Dzięki temu można również znaleźć inne problemy, np. wynikający z tego, że program nie jest uruchomiony, albo z tego, że program nie wykonał spodziewanej akcji.

Zazwyczaj wpisy w logach mają strukturę podobną do poniższej:

```
2020-04-17 10:12:45 INFO Program started.  
2020-04-17 10:20:11 ERROR Configuration file not found (/etc/program.conf)
```

Na początku linijki wypisywana jest data i godzina zdarzenia, następnie typ wpisu (np. DEBUG, INFO, WARN albo ERROR), a potem treść. Możemy dzięki temu stwierdzić, że w powyższym dzienniku wystąpił jeden błąd o godzinie 10:20.

Należy pamiętać, że nie każde dzienniki będą miały taką samą strukturę, ponieważ zależy ona od programu i może się nieco różnić. Czasami nie dodawany jest typ zdarzenia, a czasami data i godzina może mieć inny format. Niektóre programy (takie jak X.org) zamiast godziny w logach dodają czas od uruchomienia programu, a niektóre błędy



są zapisywane w oddzielnym pliku.

Najważniejsza jest treść i to ona dostarcza ważnych informacji, które przydatne są do odnalezienia przyczyny problemu.

## Przeglądanie dzienników

W systemach linuxowych większość dzienników zapisuje się w folderze `/var/log` (bądź w innym, w zależności od konfiguracji programu). Dzienniki, dla szybkiego zapisywania i łatwego przeglądania, zapisywane są w formacie tekstowym. Dzięki temu można je przeglądać za pomocą programu **less** albo **cat**.

Zazwyczaj większość dzienników w folderze `/var/log` potrzebuje uprawnień superużytkownika (np. root). Dzienniki mogą dostarczać dużo danych diagnostycznych, w tym takich, które mogą zdradzać konfigurację systemu bądź działania innych użytkowników systemu. Ze względów bezpieczeństwa dostęp do nich jest ograniczony (chyba że zostały tak skonfigurowane, aby inni użytkownicy mieli dostęp).

Podstawowym dziennikiem dostarczającym informacje o tym, co się dzieje w systemie, jest systemowy dziennik `/var/log/syslog`.

```
Apr 17 05:37:08 kali systemd[1]: Stopped User Runtime Directory /run/user/129.
Apr 17 05:37:08 kali systemd[1]: Removed slice User Slice of UID 129.
Apr 17 05:37:09 kali dbus-daemon[381]: [system] Activating via systemd: service name='org.freedesktop.UPower' unit='upower.service' requested by ':1.52' (uid=1000 pid=889 comm="xfsettingsd --display :0.0 --sm-client-id 2bad29c8")
Apr 17 05:37:10 kali systemd[1]: Starting Daemon for power management ...
Apr 17 05:37:31 kali dbus-daemon[381]: [system] Successfully activated service 'org.freedesktop.UPower'
Apr 17 05:37:31 kali systemd[1]: Started Daemon for power management.
Apr 17 05:37:52 kali dbus-daemon[769]: [session uid=1000 pid=769] Activating via systemd: service name='org.freedesktop.Notifications' unit='xfce4-notifyd.service' requested by ':1.65' (uid=1000 pid=1097 comm="/usr/lib/x86_64-linux-gnu/xfce4/panel/wrapper-2.0 ")
Apr 17 05:37:52 kali systemd[744]: Starting XFCE notifications service ...
Apr 17 05:37:53 kali dbus-daemon[769]: [session uid=1000 pid=769] Activating service name='org.freedesktop.thumbnails.Thumbnailer1' requested by ':1.67' (uid=1000 pid=1099 comm="xfdesktop --display :0.0 --sm-client-id 21b8b1e78-")
Apr 17 05:37:57 kali dbus-daemon[381]: [system] Activating via systemd: service name='org.freedesktop.ColorManager' unit='colord.service' requested by ':1.58' (uid=1000 pid=1128 comm="xiccd ")
Apr 17 05:37:57 kali dbus-daemon[769]: [session uid=1000 pid=769] Activating service name='ca.desrt.dconf' requested by ':1.73' (uid=1000 pid=1129 comm="gsettings set org.gnome.desktop.media-handling aut")
Apr 17 05:37:57 kali systemd[1]: Starting Manage, Install and Generate Color Profiles ...
Apr 17 05:37:59 kali dbus-daemon[769]: [session uid=1000 pid=769] Successfully activated service 'ca.desrt.dconf'
Apr 17 05:38:02 kali colord[1143]: failed to get edid data: EDID length is too small
Apr 17 05:38:03 kali dbus-daemon[381]: [system] Successfully activated service 'org.freedesktop.ColorManager'
Apr 17 05:38:03 kali systemd[1]: Started Manage, Install and Generate Color Profiles.
Apr 17 05:38:06 kali dbus-daemon[769]: [session uid=1000 pid=769] Successfully activated service 'org.freedesktop.Notifications'
```

Ryc. 1. Przeglądanie dzienników `/var/log/syslog` w programie `less`.

W tym dzienniku zapisywane są zdarzenia większości programów, które działają w systemie. Format logów jest trochę inny niż wyżej przedstawiano i używa poniższego wzoru:

`Apr 17 05:37:57 hostname process[PID]: Message`

Na początku zapisywana jest data w formacie miesiąc-dzień, następnie godzina, a potem nazwa maszyny (hostname). Potem zapisana jest nazwa procesu (programu), a w nawiasach kwadratowych identyfikator procesu (*process identifier* - PID). Po dwukropku wypisywany jest komunikat ze wskazanego programu.

Identyfikator procesu (PID) jest numerem, który system nadaje procesowi (programowi) w momencie uruchomienia. Może się okazać, że użytkownik uruchomił dwa programy o identycznej nazwie (np. `java`) - wtedy, aby je odróżnić, należy sprawdzić ich PID.

Dla porównania, dzienniki serwera graficznego X.org wyglądają trochę inaczej.

Zapisywane są w poniższym formacie:

```
[ 50.174] (==) Message
```

W pierwszym nawiasie kwadratowym jest czas od uruchomienia serwera (w sekundach), następnie w nawiasie okrągłym dwa znaki oznaczają marker (czyli typ, np. == oznacza zwykłą informację), a potem wypisywany jest komunikat.

Podczas przeglądania dzienników należy wziąć pod uwagę różnice w strukturze.

```
[ 50.171] Markers: (--) probed, (**) from config file, (==) default setting,
          (++) from command line, (!!) notice, (II) informational,
          (WW) warning, (EE) error, (NI) not implemented, (?) unknown.
[ 50.174] (==) Log file: "/var/log/Xorg.0.log", Time: Fri Apr 17 05:36:22 2020
[ 50.209] (==) Using system config directory "/usr/share/X11/xorg.conf.d"
[ 50.235] (==) No Layout section. Using the first Screen section.
[ 50.236] (==) No screen section available. Using defaults.
[ 50.236] (**) |→Screen "Default Screen Section" (0)
[ 50.236] (**) |   |→Monitor "<default monitor>"
[ 50.239] (==) No monitor specified for screen "Default Screen Section".
          Using a default monitor configuration.
[ 50.239] (==) Automatically adding devices
[ 50.239] (==) Automatically enabling devices
[ 50.239] (==) Automatically adding GPU devices
[ 50.240] (==) Max clients allowed: 256, resource mask: 0x1fffff
[ 50.305] (WW) The directory "/usr/share/fonts/X11/cyrillic" does not exist.
          Entry deleted from font path.
[ 50.318] (==) FontPath set to:
          /usr/share/fonts/X11/misc,
          /usr/share/fonts/X11/100dpi:unscaled,
          /usr/share/fonts/X11/75dpi:unscaled,
          /var/log/Xorg.0.log
```

Ryc. 2. Dzienniki serwera X.org.

## Wybieranie niektórych wpisów

W dużych plikach dzienników może okazać się, że ręczne „przeczesywanie” wpisów będzie bardzo trudne.

Co jeśli użytkownik chciałby zbadać dziennik `/var/log/syslog` i sprawdzić, co się stało o konkretnej godzinie, np. 6:20?

Można przeprowadzić podstawowe wyszukiwanie za pomocą programu **less**. Aby wyszukać wpisy z godziny 6:20, należy w programie less wprowadzić polecenie `/06:20` (rys. 3).

Można też dodać numery linii, aby sprawdzić, ile wpisów się znajduje w pliku, za pomocą **-N** (rys.4).

Może się pojawić potrzeba, że będziemy chcieli odfiltrować logi dla pojedynczego procesu, np. „dbus-daemon”. Niestety less nie posiada opcji filtrowania, ale możemy użyć innego programu - **grep**.

```
Apr 17 06:20:25 kali systemd[1]: Starting Daily apt upgrade and clean activities...
Apr 17 06:20:25 kali systemd[1]: Starting Clean php session files...
Apr 17 06:20:32 kali systemd[1]: phpsessionclean.service: Succeeded.
Apr 17 06:20:32 kali systemd[1]: Started Clean php session files.
Apr 17 06:20:34 kali systemd[1]: apt-daily-upgrade.service: Succeeded.
Apr 17 06:20:34 kali systemd[1]: Started Daily apt upgrade and clean activities.
Apr 17 06:25:02 kali CRON[1510]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1
1 1)
Apr 17 06:25:02 kali CRON[1511]: (root) CMD (test -x /usr/sbin/anacron || ( cd / && run-part
s --report /etc/cron.daily ))
Apr 17 06:26:35 kali kernel: [ 1844.499325] e1000: eth0 NIC Link is Down
Apr 17 06:26:37 kali kernel: [ 1846.515107] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex
, Flow Control: RX
:|
```

Ryc. 3. Efekt wyszukiwania wpisów z godziny 06:20 w programie less.

```
251 Apr 17 06:29:50 kali systemd[1]: Starting Network Manager Script Dispatcher Service
251 ..
252 Apr 17 06:29:50 kali dbus-daemon[381]: [system] Successfully activated service 'org
252 freedesktop.nm_dispatcher'
253 Apr 17 06:29:50 kali systemd[1]: Started Network Manager Script Dispatcher Service.
254 Apr 17 06:30:01 kali systemd[1]: NetworkManager-dispatcher.service: Succeeded.
255 Apr 17 06:35:02 kali CRON[1703]: (root) CMD (command -v debian-sa1 > /dev/null && d
255 bian-sa1 1 1)
(END)
```

Ryc. 4. Przeglądanie dziennika z włączonymi numerami linii.

Nazwa **grep** oznacza *global regular expression print* (drukowanie globalnych wyrażeń regularnych) i służy do wyświetlania fragmentów tekstu pasujących do wyszukiwanego wyrażenia.

Aby wyświetlić linie zawierające „dbus-daemon” z pliku `/var/log/syslog`, należy wykonać polecenie

```
grep "dbus-daemon" /var/log/syslog
```

```
root@kali:/home/user# grep "dbus-daemon" /var/log/syslog
Apr 17 05:36:42 kali dbus-daemon[381]: [system] Activating via systemd: service name='org.freedesktop
id=685 comm="/usr/bin/pulseaudio --daemonize=no ")
Apr 17 05:36:42 kali dbus-daemon[381]: [system] Successfully activated service 'org.freedesktop.Real
Apr 17 05:36:42 kali dbus-daemon[698]: [session uid=129 pid=698] Activating via systemd: service name
9 pid=692 comm="/usr/sbin/lightdm-gtk-greeter ")
Apr 17 05:36:43 kali dbus-daemon[698]: [session uid=129 pid=698] Activating via systemd: service name
129 pid=703 comm="/usr/lib/at-spi2-core/at-spi-bus-launcher ")
Apr 17 05:36:44 kali dbus-daemon[698]: [session uid=129 pid=698] Successfully activated service 'org.
Apr 17 05:36:44 kali dbus-daemon[698]: [session uid=129 pid=698] Successfully activated service 'org.
Apr 17 05:36:50 kali at-spi-bus-launcher[703]: dbus-daemon[715]: Activating service name='org.a11y.at
-gtk-greeter ")
Apr 17 05:36:50 kali at-spi-bus-launcher[703]: dbus-daemon[715]: Successfully activated service 'org.
```

Ryc. 5. Wyświetlanie linii zawierających „dbus-daemon”.

Program **grep** wyświetla wszystkie linie w terminalu bez opcji wygodnego przeglądania wyników wyszukiwania. Aby wygodnie przeglądać wyniki wyszukiwania, można przekierować je do programu **less**. Wtedy możliwe będzie przeglądanie przefiltrowanych wyników w programie **less**.

Aby to zrobić, należy wpisać następujące polecenie:

```
grep "dbus-daemon" /var/log/syslog | less
```



Znak „|” służy do przekierowania tekstu z jednego polecenia do drugiego.

```
Apr 17 05:36:42 kali dbus-daemon[381]: [system] Activating via systemd: service name='org.freedesktop.RealtimeKit1' unit='rtkit-daemon.service' requested by ':1.21' (uid=129 pid=685 comm="/usr/bin/pulseaudio --daemonize=no ")
Apr 17 05:36:42 kali dbus-daemon[381]: [system] Successfully activated service 'org.freedesktop.RealtimeKit1'
Apr 17 05:36:42 kali dbus-daemon[698]: [session uid=129 pid=698] Activating via systemd: service name='org.a11y.Bus' unit='at-spi-dbus-bus.service' requested by ':1.1' (uid=129 pid=692 comm="/usr/sbin/lightdm-gtk-greeter ")
Apr 17 05:36:43 kali dbus-daemon[698]: [session uid=129 pid=698] Activating via systemd: service name='org.gtk.vfs.Daemon' unit='gvfs-daemon.service' requested by ':1.3' (uid=129 pid=703 comm="/usr/lib/at-spi2-core/at-spi-bus-launcher ")
Apr 17 05:36:44 kali dbus-daemon[698]: [session uid=129 pid=698] Successfully activated service 'org.gtk.vfs.Daemon'
Apr 17 05:36:44 kali dbus-daemon[698]: [session uid=129 pid=698] Successfully activated service 'org.a11y.Bus'
Apr 17 05:36:50 kali at-spi-bus-launcher[703]: dbus-daemon[715]: Activating service name='org.a11y.atspi.Registry' requested by ':1.0' (uid=129 pid=692 comm="/usr/sbin/lightdm-gtk-greeter ")
Apr 17 05:36:50 kali at-spi-bus-launcher[703]: dbus-daemon[715]: Successfully activated service 'org.a11y.atspi.Registry'
:|
```

Ryc. 6. Przeglądanie wyników wyszukiwania w programie less.

## Filtrowanie logów z użyciem wyrażeń

Program grep oprócz prostego wyszukiwania frazy pozwala na użycie **wyrażeń regularnych** (ang. *regular expression*, w skrócie *regex*). Dzięki nim można podać wzorce, które mają zostać odnalezione.

Przykładowo: potrzebujemy odnaleźć wszystkie wpisy, które wystąpiły pomiędzy godziną 5:35 a 5:40. Standardowo trzeba by było wyszukać „05:35:”, „05:36:”, „05:37:”, „05:38:” oraz „05:39:”. Należałoby wykonać aż 5 różnych wyszukiwań, a do tego wyniki byłyby oddzielne i nie można by było ich przeglądać razem.

Wyrażenia regularne pozwalają sformułować taki wzorzec, który pozwoli na dopasowanie wszystkich tych godzin podczas jednego wyszukiwania.

Aby opracować taki wzorzec, należy najpierw znaleźć część wspólną dla podanych godzin - będzie to „05:3”.

Następnie zostaną podane różne możliwości dopasowania ostatniej cyfry tej godziny. Program grep powinien odnaleźć te wpisy, których godzina zaczyna się od 05:3 i kończy na cyfrze 5, 6, 7, 8 lub 9. Takie różne możliwości podaje się w nawiasach kwadratowych - [56789]. Taki zapis oznacza, że w miejscu nawiasu kwadratowego może wystąpić **jeden z podanych znaków** w tym nawiasie.

Na koniec należy wstawić dwukropek, aby nie dopasować np. godziny 12:05:35 (ponieważ bez dwukropka na końcu „05:35” w tej godzinie pasuje do wzoru).

Finalnie wyrażenie regularne będzie wyglądać następująco:

05:3[56789]:



Dzięki niemu grep znajdzie linie zawierające następujące frazy:

- 05:35:
- 05:36:
- 05:37:
- 05:38:
- 05:39:

Zapis [56789] można skrócić też do [5-9], ponieważ jest to zakres cyfr.

```
root@kali: /home/user# grep "05:3[56789]" /var/log/syslog
Apr 17 05:36:37 kali rsyslogd: [origin software="rsyslogd" swVersion="8.1911.0" x-pid="388" x-info="https://www.rsyslog.com"] rsyslogd was HUPed
Apr 17 05:36:37 kali systemd[662]: Reached target Paths.
Apr 17 05:36:37 kali systemd[662]: Reached target Timers.
Apr 17 05:36:37 kali systemd[662]: Starting D-Bus User Message Bus Socket.
Apr 17 05:36:37 kali systemd[662]: Listening on GnuPG network certificate management daemon.
Apr 17 05:36:37 kali systemd[662]: Listening on GnuPG cryptographic agent and passphrase cache (access for web browsers).
Apr 17 05:36:37 kali systemd[662]: Listening on GnuPG cryptographic agent and passphrase cache (restricted).
Apr 17 05:36:37 kali systemd[662]: Listening on GnuPG cryptographic agent (ssh-agent emulation).
Apr 17 05:36:37 kali systemd[662]: Listening on GnuPG cryptographic agent and passphrase cache.
Apr 17 05:36:37 kali systemd[662]: Listening on Sound System.
Apr 17 05:36:38 kali systemd[662]: Listening on D-Bus User Message Bus Socket.
Apr 17 05:36:38 kali systemd[662]: Reached target Sockets.
Apr 17 05:36:38 kali systemd[662]: Reached target Basic System.
Apr 17 05:36:38 kali systemd[1]: Started User Manager for UID 129.
Apr 17 05:36:38 kali systemd[662]: Starting Sound Service ...
Apr 17 05:36:38 kali systemd[1]: Started Session c1 of user lightdm.
Apr 17 05:36:38 kali systemd[1]: e2scrub_reap.service: Succeeded.
Apr 17 05:36:38 kali systemd[1]: Started Remove Stale Online ext4 Metadata Check Snapshots.
Apr 17 05:36:38 kali systemd[1]: Reached target Graphical Interface.
Apr 17 05:36:38 kali systemd[1]: Starting Update UTMP about System Runlevel Changes ...
Apr 17 05:36:39 kali systemd[1]: systemd-update-utmp-runlevel.service: Succeeded.
Apr 17 05:36:39 kali systemd[1]: Started Update UTMP about System Runlevel Changes.
```

Ryc. 7. Fragment wyników wyszukiwania z użyciem wyrażenia regularnego.

Dodatkowo, jeżeli zamiast konkretnych cyfr ([56789]) szukana fraza powinna mieć dowolny znak w danym miejscu, można użyć „.”.

05:3.:

Powyższe wyrażenie oznacza - fraza rozpoczynająca się od „05:3”, następnie jeden dowolny znak, a potem „:”.

## Rozszerzone wyrażenia regularne

Program grep domyślnie używa uproszczonych wyrażeń regularnych. Aby móc użyć rozszerzonych wyrażeń regularnych, należy użyć opcji -E.

Rozszerzone wyrażenia regularne używają pełnego silnika wyrażeń regularnych. Dzięki temu możliwe jest użycie dodatkowych wyrażeń wewnątrz zapytania.

Przykładowo możliwe jest podanie **alternatywnych** szukanych fraz. Szukając słów „failed” albo „Failed” można użyć poniższego wzorca:

failed|Failed

Pionowa linia oddziela alternatywne frazy. Co ważne, wyrażenia regularne dopuszczają użycie nawiasów do grupowania fraz - powyższy przykład można zapisać także jako poniższy wzorec:

(f|F)ailed

W tym przypadku alternatywne będą tylko te frazy, które znajdują się wewnątrz nawiasu - czyli f oraz F. To wyrażenie dopasuje więc wszystkie słowa „failed” oraz „Failed”.

To samo można też zapisać bez wykorzystania rozszerzonych wyrażeń regularnych w poniższy sposób:

`[fF]ailed`

```
root@kali:/home/user# grep -E "failed|Failed" /var/log/syslog
Apr 17 05:38:02 kali colord[1143]: failed to get edid data: EDID length is too small
Apr 17 05:38:13 kali dbus-daemon[381]: [system] Activation via systemd failed for unit 'dbus
Apr 17 05:38:18 kali dbus-daemon[381]: [system] Activation via systemd failed for unit 'dbus
Apr 17 05:38:20 kali udisksd[1201]: failed to load module mdraid: libbd_mdraid.so.2: cannot
Apr 17 05:38:20 kali udisksd[1201]: Failed to load the 'mdraid' libblockdev plugin
root@kali:/home/user# grep -E "(f|F)ailed" /var/log/syslog
Apr 17 05:38:02 kali colord[1143]: failed to get edid data: EDID length is too small
Apr 17 05:38:13 kali dbus-daemon[381]: [system] Activation via systemd failed for unit 'dbus
Apr 17 05:38:18 kali dbus-daemon[381]: [system] Activation via systemd failed for unit 'dbus
Apr 17 05:38:20 kali udisksd[1201]: failed to load module mdraid: libbd_mdraid.so.2: cannot
Apr 17 05:38:20 kali udisksd[1201]: Failed to load the 'mdraid' libblockdev plugin
root@kali:/home/user# grep "[fF]ailed" /var/log/syslog
Apr 17 05:38:02 kali colord[1143]: failed to get edid data: EDID length is too small
Apr 17 05:38:13 kali dbus-daemon[381]: [system] Activation via systemd failed for unit 'dbus
Apr 17 05:38:18 kali dbus-daemon[381]: [system] Activation via systemd failed for unit 'dbus
Apr 17 05:38:20 kali udisksd[1201]: failed to load module mdraid: libbd_mdraid.so.2: cannot
Apr 17 05:38:20 kali udisksd[1201]: Failed to load the 'mdraid' libblockdev plugin
```

Ryc. 8. Trzy sposoby wyszukiwania słów „failed” oraz „Failed”.

Dzięki rozszerzonym wyrażeniom regularnym możliwe jest wyszukanie zakresu czasu.

Przykładowo, należy wyszukać wszystkie wpisy pomiędzy godziną 05:38 a 05:42. Należy więc dopasować godziny 05:38, 05:39, 05:40, 05:41.

Na początek zostaną opracowane proste wyrażenia regularne, które pozwolą nam wyszukać przedziały 05:38-05:40 oraz 05:40-05:42.

W pierwszym przypadku będzie to „05:3[89]:”. W drugim „05:4[01]:”.

Teraz należy opracować takie wyrażenie regularne, które połączy wyniki wyszukiwania tych dwóch zapytań - zostanie użyta do tego alternatywa (znak „|”).

Wzorzec łączący te dwa wyszukiwania będzie wyglądał następująco:

`05:3[89]:|05:4[01]:`

Oprócz alternatywy, rozszerzone wyrażenia regularne pozwalają również na zdefiniowanie **ilości powtórzeń** znaków bądź grup.

Przykładowo, aby znaleźć znak „e”, który jest powtórzony 2 razy, należy poniższe wpisać wyrażenie regularne.

`e{2}`



```
root@kali:/home/user# grep -E "05:3[89]:|05:4[01]:" /var/log/syslog
Apr 17 05:38:02 kali colord[1143]: failed to get edid data: EDID length is too small
Apr 17 05:38:03 kali dbus-daemon[381]: [system] Successfully activated service 'org.freedesktop.ColorManager'
Apr 17 05:38:03 kali systemd[1]: Started Manage, Install and Generate Color Profiles.
Apr 17 05:38:06 kali dbus-daemon[769]: [session uid=1000 pid=769] Successfully activated service 'org.freedesktop.Avahi'
Apr 17 05:38:06 kali systemd[744]: Started XFCE notifications service.
Apr 17 05:38:13 kali dbus-daemon[381]: [system] Activating via systemd: service name='org.freedesktop.Avahi' is already
d=127 pid=1171 comm="/usr/lib/colord/colord-sane ")
Apr 17 05:38:13 kali dbus-daemon[381]: [system] Activation via systemd failed for unit 'dbus-org.freedesktop.Avahi'
Apr 17 05:38:14 kali NetworkManager[382]: <info> [1587116294.3800] agent-manager: req[0x55c96f8ad180, :1.7
Apr 17 05:38:18 kali dbus-daemon[381]: [system] Activating via systemd: service name='org.freedesktop.Avahi' is already
d=127 pid=1171 comm="/usr/lib/colord/colord-sane ")
Apr 17 05:38:18 kali dbus-daemon[381]: [system] Activation via systemd failed for unit 'dbus-org.freedesktop.Avahi'
Apr 17 05:38:18 kali org.freedesktop.thumbnails.Thumbnailer1[769]: Registered thumbnailer /usr/bin/gdk-pixbuf-thumbnailer
Apr 17 05:38:18 kali org.freedesktop.thumbnails.Thumbnailer1[769]: Registered thumbnailer /usr/bin/gdk-pixbuf-thumbnailer
Apr 17 05:38:18 kali dbus-daemon[769]: [session uid=1000 pid=769] Activating via systemd: service name='org.freedesktop.Avahi' requested by ':1.84' (uid=1000 pid=1119 comm="/usr/lib/x86_64-linux-gnu/tumbler-1/tumblerd ")
Apr 17 05:38:18 kali systemd[744]: Starting Virtual filesystem service - disk device monitor ...
Apr 17 05:38:19 kali dbus-daemon[381]: [system] Activating via systemd: service name='org.freedesktop.UDisks2' is already
omm="/usr/lib/gvfs/gvfs-udisks2-volume-monitor ")
Apr 17 05:38:19 kali systemd[1]: Starting Disk Manager ...
Apr 17 05:38:19 kali udisksd[1201]: udisks daemon version 2.8.4 starting
Apr 17 05:38:20 kali udisksd[1201]: failed to load module mdraid: libbd_mdraid.so.2: cannot open shared object file: No such file or directory
Apr 17 05:38:20 kali udisksd[1201]: Failed to load the 'mdraid' libblockdev plugin
Apr 17 05:38:22 kali dbus-daemon[381]: [system] Successfully activated service 'org.freedesktop.UDisks2'
Apr 17 05:38:22 kali systemd[1]: Started Disk Manager.
Apr 17 05:38:22 kali udisksd[1201]: Acquired the name org.freedesktop.UDisks2 on the system message bus
Apr 17 05:38:22 kali dbus-daemon[769]: [session uid=1000 pid=769] Successfully activated service 'org.gtk.vfs.Daemon-1.0'
Apr 17 05:38:22 kali systemd[744]: Started Virtual filesystem service - disk device monitor.
Apr 17 05:38:22 kali dbus-daemon[769]: [session uid=1000 pid=769] Successfully activated service 'org.freedesktop.Avahi'
Apr 17 05:38:23 kali dbus-daemon[769]: [session uid=1000 pid=769] Activating via systemd: service name='org.freedesktop.Avahi' requested by ':1.84' (uid=1000 pid=1099 comm="xfdesktop --display :0.0 --sm-client-id 21b8b1e78-")
Apr 17 05:38:23 kali systemd[744]: Starting Virtual filesystem metadata service ...
Apr 17 05:38:23 kali dbus-daemon[769]: [session uid=1000 pid=769] Successfully activated service 'org.gtk.vfs.Metadata-1.0'
Apr 17 05:38:23 kali systemd[744]: Started Virtual filesystem metadata service.
Apr 17 05:39:01 kali CRON[1230]: (root) CMD ( [ -x /usr/lib/php/sessionclean ] && if [ ! -d /run/systemd/sessions ]; then mkdir -p /run/systemd/sessions; fi)
Apr 17 05:39:01 kali systemd[1]: Starting Clean php session files ...
Apr 17 05:39:03 kali systemd[1]: phpsessionclean.service: Succeeded.
Apr 17 05:39:03 kali systemd[1]: Started Clean php session files.
Apr 17 05:41:55 kali systemd[1]: Started Network Manager Script Dispatcher Service.
```

Ryc. 9. Fragment wyników wyszukiwania zakresu godzin 05:38-05:42.

```
root@kali:/home/user# grep -E "e{2}" /var/log/syslog
Apr 17 05:36:38 kali systemd[1]: e2scrub_reap.service: Succeeded.
Apr 17 05:36:39 kali systemd[1]: systemd-update-utmp-runlevel.service: Succeeded.
```

Ryc. 10. Wyszukiwanie podwójnych wystąpień „e”.

Klamry służą do wpisania oczekiwanej ilości znaków (bądź grup). Powyższe wyrażenie wyszuka więc wszystkie wystąpienia „ee” (rys. 10).

W podobny sposób można wyszukać potrójne wystąpienia „e” - używając `e{3}`. Zakresy powtórzeń działają też z grupami znaków (rys. 11).

`(ed){2}`

```
root@kali:/home/user# grep -E "(ed){2}" /var/log/syslog
Apr 17 05:36:38 kali systemd[1]: e2scrub_reap.service: Succeeded.
Apr 17 05:36:39 kali systemd[1]: systemd-update-utmp-runlevel.service: Succeeded.
Apr 17 05:36:44 kali systemd[1]: NetworkManager-dispatcher.service: Succeeded.
```

Ryc. 11. Wyszukiwanie podwójnych wystąpień „ed”.



Powyższe wyrażenie wyszuka wszystkie podwójne wystąpienia „ed” (czyli „eded”)  
Ilość powtórzeń można wykorzystać też przy wyszukiwaniu godzin - przykładowo można użyć poniższego wyrażenia do wyszukania wszystkich wpisów z godziny 5 (rys. 12).

`05:[0-9]{2}:[0-9]{2}`

```
root@kali:/home/user# grep -E "05:[0-9]{2}:[0-9]{2}" /var/log/syslog
Apr 17 05:36:37 kali rsyslogd: [origin software="rsyslogd" swVersion="8.1911.0" x-pid="388" x-info="https://www.rsyslog.com"] rsyslogd was HUPed
Apr 17 05:36:37 kali systemd[662]: Reached target Paths.
Apr 17 05:36:37 kali systemd[662]: Reached target Timers.
Apr 17 05:36:37 kali systemd[662]: Starting D-Bus User Message Bus Socket.
Apr 17 05:36:37 kali systemd[662]: Listening on GnuPG network certificate management daemon.
Apr 17 05:36:37 kali systemd[662]: Listening on GnuPG cryptographic agent and passphrase cache (access for web browsers).
Apr 17 05:36:37 kali systemd[662]: Listening on GnuPG cryptographic agent and passphrase cache (restricted).
Apr 17 05:36:37 kali systemd[662]: Listening on GnuPG cryptographic agent (ssh-agent emulation).
Apr 17 05:36:37 kali systemd[662]: Listening on GnuPG cryptographic agent and passphrase cache.
Apr 17 05:36:37 kali systemd[662]: Listening on Sound System.
```

Ryc. 12. Wyszukiwanie wpisów z godziny 5.

Ilość powtórzeń pozwala też na zdefiniowanie zakresu - przykładowo aby wyszukać wszystkie linie zawierające liczby od 3 do 5 cyfr, można użyć poniższego wzorca:

`[0-9]{3,5}`

```
root@kali:/home/user# grep -E "[0-9]{3,5}" /var/log/syslog
Apr 17 05:36:37 kali rsyslogd: [origin software="rsyslogd" swVersion="8.1911.0" x-pid="388" x-info="https://www.rsyslog.com"] rsyslogd was HUPed
Apr 17 05:36:37 kali systemd[662]: Reached target Paths.
Apr 17 05:36:37 kali systemd[662]: Reached target Timers.
Apr 17 05:36:37 kali systemd[662]: Starting D-Bus User Message Bus Socket.
```

Ryc. 13. Wyszukiwanie wpisów z liczbami od 3 do 5 cyfr.

Można również połączyć te powtórzenia ze znakiem „.”, oznaczającym dowolny znak.

`(e.){3,5}`

```
root@kali:/home/user# grep -E "(e.){3,5}" /var/log/syslog
Apr 17 05:36:48 kali systemd[1]: Started Daily man-db regeneration.
Apr 17 05:51:26 kali systemd-tmpfiles[1398]: /usr/lib/tmpfiles.d/iodined.conf:1: Line references path below legacy directory /var/run/, updating /var/run/iodine -> /run/iodine; please update the tmpfiles.d/ drop-in file accordingly.
```

Ryc. 14. Wyszukiwanie wpisów z „e.” powtórzonym od 3 do 5 razy.

Powyższe wyrażenie dopasuje „e.” powtórzone od 3 do 5 razy, gdzie zamiast kropki może wystąpić dowolny znak.

## Zaawansowane opcje grep

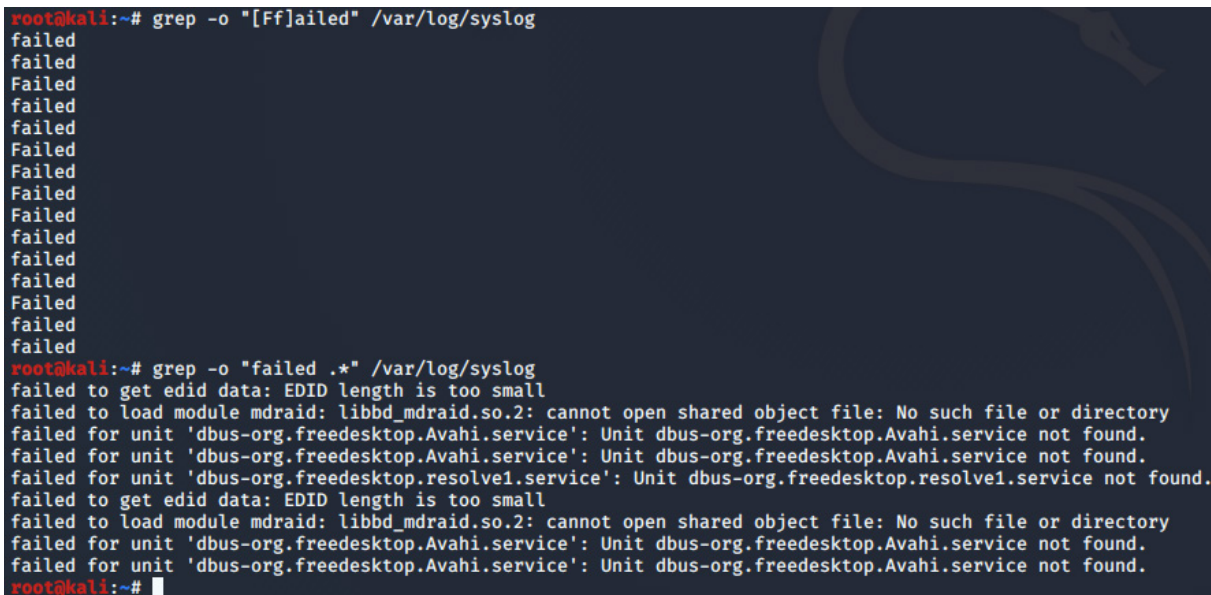
Zdarza się, że czasami jest potrzeba wyświetlenia tylko dopasowanego fragmentu tekstu.

W takich przypadkach, grep dostarcza opcji `-o`, która pozwala na wyświetlenie tylko dopasowanej części. Na przykład, aby grep pokazał tylko znalezione wystąpienia „failed” oraz „Failed”, należy użyć:

```
grep -o "[Ff]ailed" /var/log/syslog
```

Ewentualnie, żeby pokazać również to, co jest po słowie „failed”, można użyć poniższego polecenia:

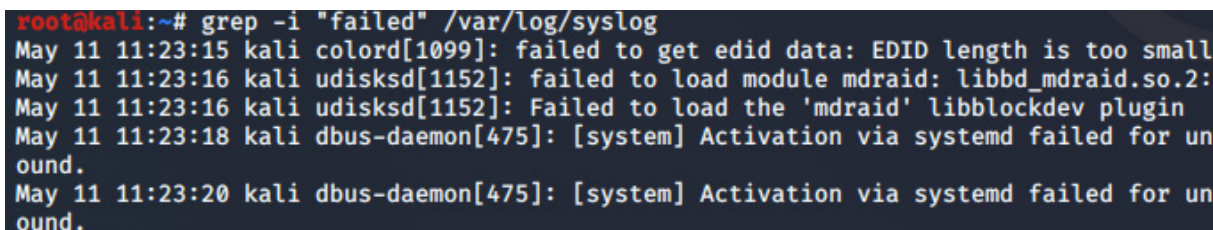
```
grep -o "failed .*" /var/log/syslog
```



```
root@kali:~# grep -o "[Ff]ailed" /var/log/syslog
failed
failed
Failed
failed
failed
Failed
Failed
Failed
Failed
failed
failed
Failed
failed
failed
failed
root@kali:~# grep -o "failed .*" /var/log/syslog
failed to get edid data: EDID length is too small
failed to load module mdraid: libbd_mdraid.so.2: cannot open shared object file: No such file or directory
failed for unit 'dbus-org.freedesktop.Avahi.service': Unit dbus-org.freedesktop.Avahi.service not found.
failed for unit 'dbus-org.freedesktop.Avahi.service': Unit dbus-org.freedesktop.Avahi.service not found.
failed for unit 'dbus-org.freedesktop.resolve1.service': Unit dbus-org.freedesktop.resolve1.service not found.
failed to get edid data: EDID length is too small
failed to load module mdraid: libbd_mdraid.so.2: cannot open shared object file: No such file or directory
failed for unit 'dbus-org.freedesktop.Avahi.service': Unit dbus-org.freedesktop.Avahi.service not found.
failed for unit 'dbus-org.freedesktop.Avahi.service': Unit dbus-org.freedesktop.Avahi.service not found.
```

Ryc. 15. Wyświetlanie tylko dopasowanej części.

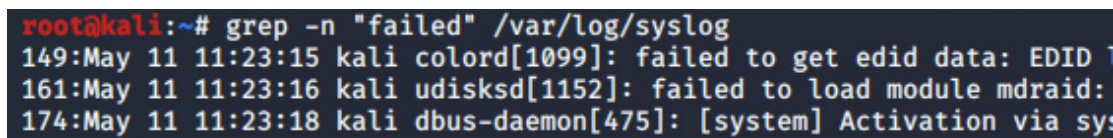
Aby uprościć wyszukiwanie „[Ff]ailed”, można również użyć opcji `-i` (rys. 16). Ta opcja włącza wyszukiwanie nieuwzględniające wielkości liter, co oznacza, że użycie „failed” dopasuje każde wystąpienie „Failed”, „fAiled”, „failed” itd.



```
root@kali:~# grep -i "failed" /var/log/syslog
May 11 11:23:15 kali colord[1099]: failed to get edid data: EDID length is too small
May 11 11:23:16 kali udisksd[1152]: failed to load module mdraid: libbd_mdraid.so.2:
May 11 11:23:16 kali udisksd[1152]: Failed to load the 'mdraid' libblockdev plugin
May 11 11:23:18 kali dbus-daemon[475]: [system] Activation via systemd failed for un
ound.
May 11 11:23:20 kali dbus-daemon[475]: [system] Activation via systemd failed for un
ound.
```

Ryc. 16. Wyszukiwanie nieuwzględniające wielkości liter.

Program `grep` może także pokazywać numery linii, w których zostały odnalezione frazy, poprzez użycie opcji `-n` (rys. 17).



```
root@kali:~# grep -n "failed" /var/log/syslog
149:May 11 11:23:15 kali colord[1099]: failed to get edid data: EDID length is too small
161:May 11 11:23:16 kali udisksd[1152]: failed to load module mdraid: libbd_mdraid.so.2:
174:May 11 11:23:18 kali dbus-daemon[475]: [system] Activation via systemd failed for un
```

Ryc. 17. Wyświetlanie linii, w których zostały odnalezione frazy.

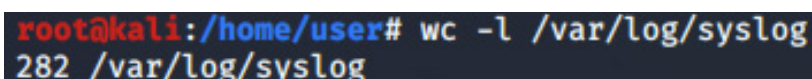
## Statystyki plików dziennika

W łatwy sposób można pobrać statystyki plików tekstowych za pomocą programu **wc**.

Nazwa **wc** oznacza *word count* (liczba słów).

Polecenie **wc** może służyć do liczenia słów (-w), znaków (-m), linii (-l) albo bajtów (-c).  
Przykładowo - aby sprawdzić liczbę linii w pliku /var/log/syslog, należy użyć polecenia:

```
wc -l /var/log/syslog
```



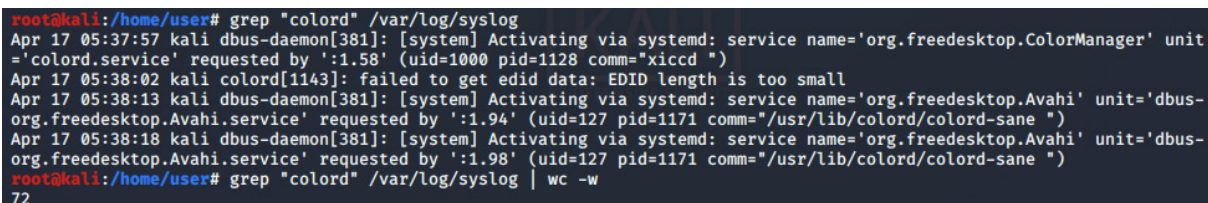
```
root@kali:/home/user# wc -l /var/log/syslog
282 /var/log/syslog
```

Ryc. 18. Liczenie liczby linii za pomocą wc.

Polecenie **wc** może również służyć do liczenia słów, znaków, linii bądź bajtów z wyniku wywołania innego polecenia. Przykładowo - można policzyć liczbę słów, które zwróci polecenie **grep "colord" /var/log/syslog**.

Aby to zrobić, należy uruchomić poniższe polecenie:

```
grep "colord" /var/log/syslog | wc -w
```



```
root@kali:/home/user# grep "colord" /var/log/syslog
Apr 17 05:37:57 kali dbus-daemon[381]: [system] Activating via systemd: service name='org.freedesktop.ColorManager' unit='colord.service' requested by ':1.58' (uid=1000 pid=1128 comm="xiccd ")
Apr 17 05:38:02 kali colord[1143]: failed to get edid data: EDID length is too small
Apr 17 05:38:13 kali dbus-daemon[381]: [system] Activating via systemd: service name='org.freedesktop.Avahi' unit='dbus-org.freedesktop.Avahi.service' requested by ':1.94' (uid=127 pid=1171 comm="/usr/lib/colord/colord-sane ")
Apr 17 05:38:18 kali dbus-daemon[381]: [system] Activating via systemd: service name='org.freedesktop.Avahi' unit='dbus-org.freedesktop.Avahi.service' requested by ':1.98' (uid=127 pid=1171 comm="/usr/lib/colord/colord-sane ")
root@kali:/home/user# grep "colord" /var/log/syslog | wc -w
72
```

Ryc. 19. Liczenie liczby wyrazów z polecenia grep za pomocą wc.



## Spis komend

**grep** [opcje] wyrażenie [plik]  
Służy do wyszukiwania wyrażenia w tekście

Opcje:

- E włącza tryb rozszerzonych wyrażeń regularnych
- o wyświetla tylko dopasowaną frazę
- i wyszukiwanie bez rozróżniania wielkości znaków
- n wyświetlanie numerów linii
- c wyświetlanie liczby znalezionych linii

**wc** [opcje] [plik]  
Wyświetla liczbę słów, znaków, linii bądź bajtów w tekście

Opcje:

- |             |                         |
|-------------|-------------------------|
| -w, --words | wyświetla liczbę słów   |
| -m, --chars | wyświetla liczbę znaków |
| -l, --lines | wyświetla liczbę linii  |
| -c, --bytes | wyświetla liczbę bajtów |