Samsung R&D Institute Poland

Linux Tech Guide

Poszerzaj i utrwalaj znajomość zagadnień technicznych, korzystając z praktycznych wskazówek naszych ekspertów.

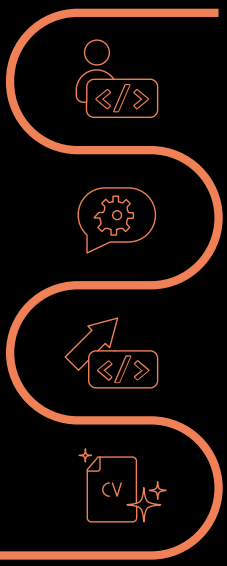
Samsung Tech Guides

Drogi kandydacie!
Droga kandydatko!

Zapraszamy Cię do zapoznania się z Linux Technical Guide - technicznym przewodnikiem poświęconym jednemu z najbardziej popularnych systemów operacyjnych!

Znajdziesz w nim tematy, które warto powtórzyć przed spotkaniem technicznym w rekrutacji do Samsung R&D!

Linux Technical Guide - Dla kogo?



dla kandydatów aplikujących na stanowiska, w których wymagana jest znajomość Linuxa,

dla kandydatów zaproszonych już na spotkanie techniczne do Samsung R&D,

dla początkujących Inżynierów chcących poszerzyć/utrwalić swoją wiedzę z Linuxa,

dla osób zainteresowanych w przyszłości rekrutacją do Samsung R&D.

Powodzenia!

Podstawy Linuxa

Podstawowym narzędziem pracy każdego programisty jest komputer wyposażony w system operacyjny, zazwyczaj jedną z dystrybucji Linuxa (lub precyzyjniej - GNU/Linux). To popularna rodzina uniksopodobnych systemów operacyjnych opartych na jądrze Linux.

Rozwijane oprogramowanie często trafia na inne systemy operacyjne oparte na tym jądrze, jak np. Tizen. W związku z tym dobrze jest znać podstawy działania tych systemów jak również narzędzia, z którymi będziesz mieć do czynienia na co dzień.

Uruchamianie Linuxa

Poznaj proces uruchamiania systemu operacyjnego od włączenia zasilania do pojawienia się okna logowania. Kilka pomocniczych pytań:

- Co to jest BIOS, UEFI i czym się różnią?
- Co to jest MBR, GPT i czym się różnią?
- Co to jest program rozruchowy (ang. bootloader)?
- Co to jest jądro systemu operacyjnego?

Dobrze jest umieć rozwinąć powyższe akronimy.
W przypadku jądra dobrze jest wiedzieć, jakie moduły są przez nie inicjalizowane oraz jaka jest ich rola i działanie.

Powłoka systemowa

Powłoka systemowa (ang. shell) to podstawowy interfejs użytkownika pozwalający na interakcję z systemem operacyjnym. Przy pomocy powłoki można wywoływać polecenia i oglądać ich rezultat na ekranie. Często konieczne jest pisanie skryptów, dzięki którym można zautomatyzować wiele zadań. Znajomość działania powłoki jest bardzo przydatna w codziennej pracy. W tym dokumencie zakładamy, że omawianą powłoką jest `bash`. Poniższe pytania mają zwrócić Twoją uwagę na najważniejsze kwestie:

- Każda komenda zwraca jakiś wynik.
  Jak sygnalizowany jest sukces, a jak błąd?
- Jak wykonać polecenie w tle?
- Jak wywołać kilka poleceń w jednej linii, jedno po drugim? Jak wywołać polecenie, jeżeli poprzednie zakończyło się sukcesem/błędem? Rozważ operatory `;` (średnik), `&&` (dwa znaki handlowe `i`, ang. et lub ampersand) i `||` (dwie kreski pionowe).
- Jak napisać skrypt powłoki? Jak deklaruje i definiuje się zmienne, jak używa się ich w poleceniach?

Powłoka systemowa

Argumentami poleceń na ogół są pliki. Co warto wiedzieć a propos nazw i ścieżek do pliku:

- Czym są i czym różnią się względne i bezwzględne ścieżki do plików?
- W tworzeniu ścieżek pomagają znaki `~` (tylda), `.` (kropka) i `..` (dwie kropki). Co oznaczają?
- Co to jest mechanizm ekspansji nazw plików (ang. globbing)? Co robią symbole wieloznaczne `?` (znak zapytania) oraz `*` (asterysk)?
  Jak w ścieżce/nazwie pliku podać, że możliwe jest wystąpienie pewnych znaków (nawiasy kwadratowe `[` , `]`)?

Niektóre znaki mają specjalne znaczenie, jak chociażby wyżej wymieniony `*`. Można ich użyć w poleceniu dosłownie, poprzedzając je znakiem ucieczki, którym w powłoce jest `\` (ukośnik wsteczny, ang. backslash). To działanie nazywamy po angielsku „escaping". Jeżeli w nazwie mamy dużo znaków specjalnych, to możemy użyć ich wszystkich dosłownie otaczając całą nazwę apostrofami (znak `'`). Możliwe jest też użycie cudzysłowów (znak `"`), które zachowują dosłowne znaczenie znaków wewnątrz oprócz `\`, `$` - referencja do zmiennej i `` ` `` - grawis służący do podstawienia komendy (ang. command substitution - dowiedz się, co to).

W codziennej pracy przydaje się też zdalny dostęp do powłoki przy pomocy narzędzia `ssh`. Możliwe jest też kopiowanie plików przez sieć przy pomocy `scp`.
Zapoznaj się z nimi.

Podstawowe polecenia

Ważne jest, aby rozumieć i znać polecenia, których się używa. Na ogół komendy posiadają wbudowany poradnik, który można wyświetlić wywołując je z argumentem `--help`. Przydatny jest też podręcznik systemowy (ang. manual), który można czytać przy pomocy polecenia `man`. Warto opanować przeglądanie stron podręcznika, a w szczególności wybieranie konkretnych rozdziałów (ang. section) w przypadku występowania hasła w kilku rozdziałach.
Naucz się korzystać z podstawowych poleceń:

- Wyświetlanie plików: `echo`, `cat`, `more`, `less`, `head`, `tail`
- Edycja tekstu: `nano`, `vim` (jak wyjść z `vim`?)
- Nawigacja między katalogami: `pwd`, `ls`, `cd`
- Zarządzanie plikami i katalogami: `cp`, `mv`, `mkdir`, `touch`, `rm`, `dd`
- Tworzenie dowiązań (ang. link) twardych i symbolicznych (czym się różnią?): `ln`
- Porównywanie plików i nakładanie zmian: `diff`, `patch`
- Praca z archiwami: `tar`
- Szukanie plików: `find`

Zmienne środowiskowe

W systemie operacyjnym zdefiniowany jest szereg zmiennych wpływających na działanie programów i procesów. Nazywamy je zmiennymi środowiskowymi. Jedną z nich jest zmienna `PATH`, która zawiera listę katalogów, w których powłoka szuka plików wykonywalnych. Innym przykładem jest `HOME`, która wskazuje katalog domowy użytkownika.

- Jak wypisać na ekran wszystkie zmienne środowiskowe?
- Jak wyświetlić wartość konkretnej zmiennej środowiskowej?
- Jak zmienić zmienną środowiskową?
- Jak dodać nową zmienną środowiskową?
- Jak zachować dodaną zmienną środowiskową między sesjami?

Pliki i systemy plików

Jedną z podstawowych koncepcji systemów uniksopodobnych jest traktowanie wszystkich obiektów jako plików (chyba że to procesy). Tyczy się to nie tylko zwykłych plików, ale też katalogów, urządzeń blokowych i znakowych, dowiązań symbolicznych, potoków (ang. pipe), czy gniazdek (ang. socket). Informacje o plikach przechowywane są w strukturach o nazwie i-węzeł (ang i-node). Dane w nich zawarte można odczytać poleceniem `stat`.

- Czym są wymienione wyżej rodzaje plików?
- Co to są ukryte pliki? Jak utworzyć ukryty plik?
- Co robią urządzenia znakowe `/dev/null`, `/dev/random`, `/dev/urandom`, `/dev/zero`?

Pliki przechowywane są w systemach plików. Istnieje wiele systemów plików, które różnią się wydajnością, stabilnością, bezpieczeństwem i innymi cechami. Jednym z najczęściej używanych we współczesnych Linuxach jest `ext4`. Drzewo systemu plików zaczyna się w katalogu głównym `/`. Część tej struktury jest ustandaryzowana przez normę POSIX, tak więc w każdym systemie można spodziewać się pewnych podkatalogów przeznaczonych do konkretnych celów.

- Jakie są systemy plików? Czym się charakteryzują, zwłaszcza `ext4`?
- Co to jest dziennikujący (księgujący, ang. journaling) system plików?
- Jakie podstawowe podkatalogi znajdują się w katalogu głównym?
- Poznaj przydatne polecenia: `mkfs`, `mount`, `umount`, `df`, `fdisk`, `fsck`.

Użytkownicy i grupy, prawa dostępu

Użytkownicy, grupy i prawa dostępu to podstawowe mechanizmy pozwalające na kontrolowanie dostępu do plików i procesów. Systemy uniksopodobne zaprojektowane są tak, aby umożliwić wielodostępność (wielu użytkowników może korzystać z systemu) i wielozadaniowość (użytkownik może uruchamiać wiele zadań równolegle). W związku z tym istotną kwestią jest zapewnienie, że użytkownik nie będzie mieć dostępu do plików, do których nie powinien. Istnieje specjalne konto, które ma pełną kontrolę nad systemem operacyjnym – `root` (lub superuser). Pozostali użytkownicy mogą wykonywać pewne operacje w zależności od przynależności do grup oraz praw dostępu plików.

Co należy wiedzieć:

- Jak zarządzać użytkownikami? Jak zmieniać im hasło?
- Czym są grupy? Jak nimi zarządzać?
- Jak działa mechanizm praw dostępu do pliku? Jak wyświetlić prawa dostępu (polecenie `ls`)? Jak je zmienić (polecenie `chmod`)?
- Jak wykonywać polecenia jako inny użytkownik (polecenie `su`)?
- Jak zmienić właściciela pliku (polecenia `chown`, `chgrp`)?

Strumienie

Strumienie to interfejs systemów uniksopodobnych służący do przesyłania uporządkowanych bajtów w jednym kierunku. W ten sposób użytkownik może komunikować się z terminalem, terminal z procesami, a procesy między sobą. Strumienie służą między innymi do czytania i pisania do plików. Każdy proces ma domyślnie przydzielone i otwarte trzy strumienie do komunikacji z powłoką:

- `stdin` – standardowe wejście (identyfikator 0)
- `stdout` – standardowe wyjście (identyfikator 1)
- `stderr` – standardowe wyjście błędów (identyfikator 2)

Strumienie można przekierować do/z plików oraz programów. Służą do tego operatory:

- `>` – nadpisanie pliku zawartością strumienia
- `>>` – dopisanie zawartości strumienia na koniec pliku
- `<` – przekierowanie strumienia wejściowego

Strumienie

Można przekierować wybrany strumień przy pomocy identyfikatora lub zignorować go przekierowując do specjalnego urządzenia `dev/null`. Można też połączyć strumienie, aby zarówno standardowe wyjście, jak i standardowe wyjście błędów trafiały do tego samego miejsca (konstrukcja `2>&1`). Naucz się korzystać z tych operacji. Zapoznaj się też z dokumentami wbudowanymi (ang. heredoc), które często przydają się w skryptach.

Kolejnym sposobem przekierowania strumienia z jednego procesu do drugiego jest utworzenie potoku wykonawczego przy pomocy operatora `|` (kreska pionowa, ang. pipe). Operator ten pozwala łączyć wyjście jednego programu z wejściem drugiego. To pozwala na tworzenie skomplikowanych operacji przy pomocy łączenia ze sobą (potokowania) prostych poleceń. Opanuj ten mechanizm.

Dzięki przekierowaniu strumieni i potokowaniu możliwe jest zaawansowane przetwarzanie danych z poziomu linii poleceń lub w skryptach powłoki. W systemach GNU/Linux dostępnych jest wiele narzędzi potrafiących przetwarzać strumień wejściowy zamiast pliku - nazywamy takie narzędzia filtrami. Część filtrów związanych z wyświetlaniem plików była wymieniona już wyżej.

Kilka innych, przykładowych narzędzi (jest ich więcej!). Naucz się z nich korzystać:

- `cut`, `join`, `paste` - przetwarzanie danych rozdzielanych separatorami
- `wc` - zliczanie linii, słów, bajtów
- `sort` - sortowanie
- `awk`, `grep`, `sed` - zaawansowane wyszukiwanie i przetwarzanie tekstu

Procesy i sygnały

W systemach uniksopodobnych proces to obiekt wykonujący pewne instrukcje, posiadający przydzieloną pamięć, czas procesora i inne zasoby. Uruchamiając program, system operacyjny tworzy proces z całym jego środowiskiem pracy i nadaje mu identyfikator nazywany `PID`. Pierwszym procesem startującym po inicjalizacji jądra jest `init` o `PID = 1`. Wszystkie inne procesy są jego potomkami. Informacje o procesach można znaleźć w systemie plików `procfs` zamontowanym w katalogu `/proc`.

Z procesami można komunikować się przy pomocy sygnałów, które są bardzo proste - nie przekazują żadnych danych. Sygnały można blokować, ignorować lub przechwytywać. Wyjątkiem są sygnały `SIGKILL` i `SIGSTOP`, które zabijają lub zatrzymują proces.
Co trzeba wiedzieć:

- Jak wyświetlić działające procesy (polecenie `ps`)?
- Jak wyszukać interesujący nas proces (polecenie `pgrep` lub polecenie `ps` plus strumieniowe przetwarzanie tekstu, np. `grep`)?
- Jakie są sygnały? Jak je wysyłać?
- Jak zabić proces (polecenie `kill`)?
- Jak przechwycić i obsłużyć sygnał we własnym programie?

Na koniec bardzo ważne narzędzie: `strace`. Pozwala prześledzić wszystkie wywołania systemowe procesu. To bardzo przydatne, gdy szukamy powodów nieprawidłowego działania programów.

Usługi

Usługi (lub demony, ang. daemon) to procesy działające w tle, wykonujące pewne operacje zapewniające prawidłowe działanie systemu operacyjnego.
Z technicznego punktu widzenia są to procesy, które nie są podłączone do terminala i których rodzicem jest proces `init`. Sam `init` może mieć wiele implementacji. Jedne z popularniejszych to `sysvinit` i `systemd` (który oprócz implementacji `init` zawiera wiele innych narzędzi).

Usługami na ogół zarządza się przy pomocy skryptów pozwalających na uruchamianie, zatrzymywanie, restartowanie i sprawdzanie stanu. Poczytaj o wyżej wymienionych implementacjach i naucz się, jak zarządzać w nich usługami.

Rekomendowane materiały online

Warto zapoznać się z oficjalną stroną projektu GNU:

https://www.gnu.org/.

W szczególności polecamy podręcznik użytkownika GNU Bash, w którym możesz przeczytać o większości poruszanych w niniejszym dokumencie kwestiach w szczegółach:

https://www.gnu.org/software/bash/manual/.

Samsung R&D Institute Poland

Samsung Tech Guides