Systemy operacyjne – studia stacjonarne 2023/24

Lab 2. System plikowy - przeszukiwanie systemu plików, dowiązania do plików - linki, prawa dostępu.

1. Przeszukiwanie systemu plików.

Do przeglądania linuksowego systemu plików w poszukiwaniu plików służą polecenia systemowe: **grep** i **find**. Jeśli poszukiwane są pliki według ich atrybutów, zawartych w i-węźle pliku, a więc takich jak np. rozmiar, nazwa, prawa dostępu, właściciel należy zastosować polecenie find. W przypadku poszukiwania lub przeszukiwania plików ze względu na ich zawartość używamy polecenia grep. Niektóre polecenia złożone mogą wymagać zastosowania obu komend.

Polecenie grep jest uniwersalnym programem przeznaczonym do wyszukiwania w pliku wierszy zawierających określony wzorzec. Dane ze strumienia wejściowego lub z pliku wejściowego czytane wierszami i wypisywane są te wiersze, które zawierają podany wzorzec.

grep opcje wzorzec plik
Uniwersalność programu grep bierze się z możliwości tworzenia złożonych wzorców za pomocą znaków uogólniających. Znaczenie wybranych metaznaków:

. – jeden dowolny znak
***** – zero lub więcej powtórzeń poprzedzającego elementu (znaku)
.* – dowolny ciąg znaków
[a-z] – jeden znak z podanego zakresu, (lub listy np.:[abcdefghijk], [0-9a-z])

Można również określić położenie wzorca w wierszu:

^wzorzec – pasuje do linii rozpoczynających się danym wzorcem
wzorzec$ – pasuje do linii kończących się danym wzorcem
Znak ^ po otwierającym nawiasie kwadratowym pełni rolę zaprzeczenia. Np**.** [^a-z] oznacza znak nie będący małą literą.

Przykłady zastosowania wzorców polecenia grep:

abc – wypisuje wiersze zawierające łańcuch „abc”
^abc – wypisuje wiersze zaczynające się łańcuchem „abc”
abc$ – wypisuje wiersze kończące się łańcuchem „abc”
a..c – wypisuje wiersze zawierające znaki „a” i „c”, przedzielone dwoma dowolnymi znakami,
a*c – wypisuje wiersze zawierające znak „a” występujący dowolną liczbę razy i znak „c”, lub tylko znak c
a+c – wypisuje wiersze zawierające znak „a” występujący dowolną liczbę razy i znak „c”
^[Cc] – wypisuje wiersze rozpoczynające się od znaku C lub c,
^[^Cc] – wypisuje wiersze nie rozpoczynające się od znaku C ani c

Dostępne opcje:

-i – ignoruje wielkość liter przy porównywaniu tekstu ze wzorcem,
-v – wypisuje te linie, które nie zawierają podanego wzorca,
-c – podaje tylko liczbę wierszy odpowiadających wzorcowi,
-n – przed każdą linijką odpowiadającą wzorcowi podaje jej numer,
-r – umożliwia przeglądanie rekurencyjne katalogu.

Jeśli wzorzec zawiera odstępy lub znaki specjalne, należy ująć go w apostrofy lub cudzysłów. Jeśli znak specjalny ma być użyty we wzorcu jako znak zwykły, wtedy należy poprzedzić go znakiem \.
Jeśli szukany wzorzec jest alternatywą, należy użyć operatora |, który maskowany jest znakiem \ (\|).

Ćwiczenia 1:

Utwórz plik tekst następującej treści:

System operacyjny LINUX
to system wielodostępny.
System ten jest również
wielozadaniowy.

Zadania 1-10 dotyczą przeszukiwania tekst.

1. Wyświetl wszystkie linijki zawierające 'system' przy rozróżnieniu dużych i małych liter.
2. Wyświetl wszystkie linijki zawierające '**system**' ignorując wielkość liter.
3. Ile linijek zawiera wyraz '**system**' (ignorując wielkość liter)?
4. Wyświetl wszystkie linijki nie zawierające '**system**' ignorując wielkość liter.
5. Wyświetl wszystkie linijki zaczynające się od '**Sy**'.
6. Wyświetl wszystkie linijki kończące się na 'y'.
7. Wyświetl wszystkie linijki, w których '**oper**' poprzedza '**Sys**'.
8. Wyświetl te linie, które zawierają słowo '**to**' lub '**ten**'.
9. Wyświetl te linie, które zawierają zarówno słowo '**to**' jak i '**ten**' (w dowolnej kolejności).
10. Odszukaj w katalogu /usr/include w plikach nagłówkowych (pliki z rozszerzeniem h) ciąg znaków '**pow**'.
11. Wyświetl linie zawierające tekst: '**main'** ze wszystkich plików znajdujących się w poddrzewie katalogowym rozpoczynającym się w katalogu kat.
12. Przeanalizuj następujące polecenia:

```
grep ^[^d-] plik
grep "...x" plik
grep –v "^[cC]" plik.f > plikbk.f
ls –la ..|grep user1
grep "\.$" plik
grep ".$" plik
grep "int\|double" *.c
```

Do wyszukiwania plików w systemie plikowym według różnych kryteriów związanych z atrybutami pliku służą polecenia: find, whereis, which. Polecenie:

whereis plik
wyświetla ścieżki dostępu do plików o podanej nazwie, przeszukując standardowe katalogi linuksowe. Służy głównie do wyszukiwania programów w wersji źródłowej lub binarnej oraz dokumentacji do nich. Polecenie:

which plik
podaje ścieżkę dostępu do pliku, który jest wykonywany po wydaniu polecenia wskazanego przez parametr, odwołując się do zmiennej środowiskowej $PATH.
Powyższe polecenia umożliwiają wyszukiwanie plików według nazwy. Bardziej uniwersalne jest polecenie find, które umożliwia wyszukiwanie plików wg różnych kryteriów:

find katalog_startowy_szukania opcje kryterium
Najczęściej używane opcje i odpowiadające im kryteria poszukiwania:
-name – nazwa pliku
-path – nazwa ścieżkowa pliku
-type – typ; wymagany jest jednoznakowy argument, określający typ:

d – katalog
f – plik zwykły
b – plik specjalny blokowy
c – plik specjalny znakowy
s – gniazdo
p – potok
l – link symboliczny

-size – rozmiaru pliku w: blokach – b, znakach – c, słowach – w lub kilobajtach – k
np. –size +100c – pliki o rozmiarze większym niż 100 znaków
–size –100w – pliki o rozmiarze mniejszym niż 100 słów

-mtime – czas modyfikacji - liczba dni, jakie minęły od ostatniej modyfikacji
np. –mtime +3 – pliki modyfikowane więcej niż 3 dni temu
–mtime -3 – pliki modyfikowane mniej niż 3 dni temu

-atime – czas dostępu - liczba dni, jaka minęła od ostatniego dostępu

-user – użytkownik (właściciel)

-maxdepth – głębokość przeszukiwania - liczba n>0 - schodzi najwyżej n poziomów poniżej punktu początkowego

-perm – prawa dostępu – podane symbolicznie lub w formie 3 lub 4 cyfr z przedziału <0-7> Polecenie -perm 100 – pozwala na odnalezienie plików, które mają ustawione przynajmniej prawo x dla właściciela.

-exec – powoduje wykonanie na odnalezionych plikach polecenia podanego po opcji exec. Polecenie to musi być zakończone ciągiem znaków {} \;
np. find / -name ”.c” -exec cat {} \;

-ok – działa analogicznie do opcji exec, wymaga jednak dodatkowo potwierdzenia wykonania polecenia na każdym pliku

-newer – czas modyfikacji pliku późniejszy niż wskazanego pliku

Przy poszukiwaniu można korzystać również z operatorów logicznych OR, NOT, AND. Operator NOT polecenia **find** zapisuje się w postaci znaku wykrzyknika **!**. Wykrzyknik umieszczony przed kryterium poszukiwania neguje to kryterium, np. polecenie **find . ! -name ‘at’** pozwala odszukać wszystkie pliki mające nazwy różne od **at**. Operator logiczny OR zapisujemy –**o** , natomiast operator AND jako **–a,** np. polecenie **find . -name ‘at’ –o –type d** pozwala na wyszukanie plików o nazwie at lub typie **d**. Jeśli plik spełnia jedno lub drugie kryterium, jest uznawany za pasujący do wzorca. Kiedy kilka opcji zostanie podanych w wierszu poleceń, tworzą one operację AND.

Operatory **( )** wymuszają pierwszeństwo (np. jak w wyrażeniach arytmetycznych) - pozwalają tworzyć bardziej złożone zapytania.

Polecenie **find / -name ‘at’ >p1** rozpoczyna poszukiwania od głównego katalogu i szuka plików o nazwie at, a następnie zapisuje rezultat poszukiwań w pliku p1 (pełną nazwę każdego znalezionego pliku).

Ćwiczenia 2:

1. Napisz polecenie, które odnajdzie w plikach źródłowych w „C” (o rozszerzeniu c), znajdujących się w katalogu /home/inf-19/sostudent/ linie zawierające tekst: **tablica,** zapisze je w pliku tablica, w katalogu domowym, a na ekranie wyświetli liczbę znalezionych linii.
2. Napisz polecenie zliczające w całym systemie plikowym pliki, których jesteś właścicielem, a które nie znajdują się w Twoim katalogu domowym. na ekranie nie mogą pojawić się komunikaty o błędach.
3. Znajdź w systemie plikowym pliki zwykłe, modyfikowane ponad tydzień temu, do których są ustawione rozszerzone prawa dostępu na poziomie właściciela (s zamiast x) oraz katalogi z ustawionym bitem lepkości (t na poziomie reszty). Nazwy odnalezionych plików powinny

pojawić się na ekranie i zostać zapisane do pliku wynik. Na ekranie nie mogą się pojawić komunikaty o błędach.

4. Napisz polecenie zliczające w Twoim katalogu domowym (bez przeszukiwania w głąb) linie zawierające tekst „exec” w plikach zwykłych, których rozmiar jest mniejszy niż 200 słów.
5. Napisz polecenie zliczające pliki, których nazwy rozpoczynają się na literę „a” lub ”A” z katalogu głównego oraz z Twojego katalogu domowego (suma).
6. Ze swojego katalogu domowego wybierz pliki źródłowe w języku C, połącz je w jeden plik o nazwie razem, a do pliku program zapisz te linie, które nie są komentarzem (nie rozpoczynają się znakami: //.
7. W plikach z Twojego katalogu domowego, których nazwy rozpoczynają się na literę a i zawierają dowolną cyfrę (nazwy!!!) znajdź linie rozpoczynające się dowolną dużą literą lub kończące się kropką. Znalezione linie zapisz do pliku linie.

2. Adresowanie bloków dyskowych

Z punktu widzenia użytkownika plik jest identyfikowany przez nazwę ścieżkową. Na jej podstawie system operacyjny musi zlokalizować bloki dyskowe, czyli miejsce, w którym plik jest przechowywany na dysku. Pliki zwykłe zawierają dane, natomiast katalogi to pliki binarne zawierające listę plików (w tym także innych katalogów), które się w nim znajdują. Każda pozycja w katalogu opisuje jeden plik i stanowi parę informacji złożoną z nazwy pliku oraz jego numeru i-węzła. Jest to mechanizm spajający plik opisywany przez dany i-węzeł z jego położeniem w drzewie katalogowym. Sam plik nie zawiera więc żadnych informacji o tym, w jakim miejscu drzewa katalogowego jest zlokalizowany, a w katalogu nie są zawarte żadne informacje na temat atrybutów pliku (przeciwnie niż w systemach plikowych opartych na tablicy FAT). Wszystkie atrybuty pliku oraz adresy bloków danych znajdują się w i-węźle pliku, który jest podstawą lokalizacji bloków dyskowych. Wszystkie i-węzły mają unikalne numery (w ramach jednego fizycznego urządzenia w jednym systemie plikowym). W i-węźle przechowywane są następujące informacje o pliku:

- typ pliku. W Linuxie występują następujące typy plików:
  - **f** – plik zwykły,
  - **d** – katalog,
  - **p** – łącze nazwane FIFO,
  - **b** – plik specjalny blokowy,
  - **c** – plik specjalny znakowy,
  - **l** – link symboliczny,
  - **s** – gniazdo,
- identyfikator właściciela oraz grupy pliku,
- prawa dostępu,
- rozmiar pliku w bajtach,
- ostatni czas dostępu, modyfikacji,
- czas utworzenia i skasowania,
- liczba dowiązań,
- liczba bloków dyskowych zajmowanych przez plik,
- adresy dyskowe.

Systemy plikowe ext2, ext3.

Adresy dyskowe umieszczone w i-węźle wskazują na bloki danych. W i-węźle przechowywanych jest 12 adresów bezpośrednich, jeden adres bloku pojedynczo pośredniego oraz po jednym adresie dla bloków podwójnie i potrójnie pośrednich.

Chcąc odczytać dane z pliku jądro najpierw musimy odczytać i-węzeł pliku, a dopiero później bloki dyskowe z danymi. Do przechowywania adresów bloków dyskowych służy tablica składająca się z 15

elementów, która jest umieszczona w i-węźle. Pierwszych 12 adresów to adresy (numery) bloków bezpośrednich. Zakładając, że rozmiar bloku wynosi 1024 bajty, przez takie adresowanie możemy mieć dostęp bezpośrednio do 12 · 1024 (rozmiar bloku) danych. Co należy zrobić, gdy plik ma np. 100KB. Jak zaadresować kolejne części pliku? Służą do tego pozostałe trzy wpisy w tablicy adresów. Rekord 13 jest adresem bloku pojedynczo pośredniego, następne dwa rekordy to adresy bloków podwójnie i potrójnie pośrednich.

System plikowy ext4.

Wprowadzono tzw. extenty. *Za pomocą mechanizmu extents zmieniona została filozofia organizacji bloków, w których przechowywany jest plik na dysku. W systemie ext4 pliki przechowywane są w ciągłym zbiorze bloków, nazywanym extent. W i-węźle przechowywane mogą być maksymalnie 4 informacje o extent, a każdy taki zbiór może zawierać maksymalnie 128 MiB przy rozmiarze bloku 4kiB (przy większym rozmiarze pliku używane jest pośrednie adresowanie). Dane te są przechowywane zamiast wskaźników do bloków.*

3. Dowiązania do plików - linki.

Ważną i pożyteczną cechą systemu plikowego są linki. Umożliwiają one nadanie wielu nazw jednemu plikowi. Rozróżniamy dwa rodzaje dowiązań:

- linki twarde, **ln oryginalna nazwa dodatkowa nazwa**
- linki symboliczne. **ln - s ścieżka/nazwa nazwa łącznika**

W różnych częściach systemu możemy utworzyć linki, które będą wskazywały na jeden plik. Nie musimy w ten sposób tworzyć wielu kopii tego samego pliku i możemy zaoszczędzić miejsce na dysku. Linki twarde i symboliczne pełnią podobne funkcje, lecz różny jest mechanizm ich działania i nie zawsze mogą być one stosowane wymiennie.

Linki twarde (łączniki utworzone bez parametru **s** tworzą tzw. łączniki sztywne) umożliwiają tworzenie dwóch lub więcej nazw dla jednego i-węzła. Dotyczą one tego samego fizycznego pliku dyskowego, ale każdy z nich stanowi oddzielne wejście w odpowiednich katalogach. Nie zadziałają one, kiedy próbujemy łączyć plik z plikiem w katalogu innych użytkowników, który jest zlokalizowany w innym systemie plików (system plików mogą tworzyć dowolne urządzenia pamięci fizycznych).

Aby pokonać to ograniczenie, należy używać linków symbolicznych. Łącznik symboliczny przechowuje ścieżkę do pliku, z którym jest połączony, dzięki czemu może przekraczać fizyczne granice urządzeń. Nie jest to bezpośredni łącznik sztywny, ale raczej informacja o tym, jak zlokalizować konkretny plik.

Ćwiczenia 3 – *linki twarde*

1. Wykonaj komendy: **ln plik lintw** oraz **ln lintw lintw3**:
   - porównaj numery i-węzłów plików: **plik**, **lintw**, **lintw2**,
   - porównaj wszystkie atrybuty plików,
   - sprawdź jak zmieniła się ilość dowiązań w plikach,
   - porównaj zawartość wszystkich plików,
   - zmień zawartość jednego z nich i sprawdź jak wygląda zawartość pozostałych,
   - usuń plik o nazwie plik i zaobserwuj jak zmienia się ilość dowiązań w pozostałych plikach.
2. Sprawdź numery i-węzłów plików w komendach **mv** i **cp**:
   - odczytaj numer węzła wybranego pliku, np. **plik**,
   - wykonaj komendę **mv plik zmiana,**
   - odczytaj numer i-węzła pliku **zmiana**,
   - wykonaj komendę **cp plik kopia,**

- odczytaj numery i-węzłów obu plików.

Ćwiczenia 4 *– linki symboliczne*

1. Wykonaj komendę **ln –s plik linksymb**:
   - porównaj atrybuty obu plików (typ, data, rozmiar, nazwa),
   - porównaj numery i-węzłów obu plików,
   - porównaj zawartości obu plików,
   - zmień plik **linksymb** (dodaj coś do niego),
   - sprawdź, czy zmienił się również **plik**,
   - sprawdź czy zmieniły się rozmiary obu plików,
   - usuń plik **plik**,
   - wylistuj zawartość pliku **linksymb**,
   - utwórz inny plik w tym samym katalogu o nazwie **plik**,
   - sprawdź, czy ma inny numer i-węzła,
   - wylistuj plik **linksymb**.
2. Wykonaj komendę **ln –s /home drzwi**:
   - sprawdź atrybuty pliku **drzwi**,
   - zastanów się, jaki powinien być efekt wykonania komendy **ls drzwi**, a następnie sprawdź, czy miałeś rację,
   - zastanów się, jaki powinien być efekt wykonania komendy **ls -l drzwi**, a następnie sprawdź, czy miałeś rację,
   - wykonaj komendę **cd drzwi**, a następnie **ls**,
   - sprawdź komendą **pwd** bieżącą ścieżkę.

4. Prawa dostępu do plików i katalogów.

Prawa dostępu do plików

Każdy plik i katalog posiada zestaw praw dostępu określających, kto ma dostęp do pliku i jakie ma prawa. Można je wyświetlić komendą **ls –l.** Istnieją trzy kategorie użytkowników:

- właściciel pliku,
- grupa do której należy właściciel,
- pozostali użytkownicy systemu.

Prawa dostępu można nadawać plikom i katalogom. Każdy plik ma ściśle określone prawa dostępu stwierdzające, czy określony użytkownik jest uprawniony do odczytania lub zapisania pliku bądź do jego wykonania. Każdy użytkownik może mieć dowolną kombinację tych praw. Są one całkowicie niezależne i posiadanie jakiegokolwiek z nich nie jest warunkiem posiadania innego. W przypadku pliku prawa są interpretowane w następujący sposób:

- **r** – prawo czytania umożliwia oglądanie zawartości pliku, oznacza jednocześnie prawo do kopiowania,
- **w** – prawo pisania oznacza zezwolenie na modyfikację zawartości pliku,
- **x** – prawo do uruchomienia pliku wykonywalnego.

Prawa dostępu do katalogów

Te same kategorie praw **–** czytania, pisania i wykonywania odnoszą się do katalogów:

- **r** – prawo czytania umożliwia przeszukiwanie zawartości katalogu, jest interpretowane jako prawo wypisywania zawartości (komenda **ls**),
- **w** – prawo pisania daje możliwość modyfikowania zawartości katalogu, umożliwia dodawanie nowych oraz usuwanie dotychczasowych plików z katalogu,
- **x** – prawo wykonywania w stosunku do katalogu pozwala na dostęp do plików zapisanych w nim oraz na wejście do danego katalogu **–** uczynienie go katalogiem bieżącym (komenda **cd**

katalog).

Prawa dostępu konieczne do wykonania poleceń zawarte są w poniższej tabelce.

WYKONYWANE CZYNNOŚCI	**PLIK**	**KATALOG**
	r w x	**r w x**
wejście do katalogu komenda **cd**	**- - -**	**- - x**
przeglądanie, jakie pliki znajdują się w katalogu komenda **ls**	**- - -**	**r- -**
przeglądanie, jakie pliki znajdują się w katalogu i jakie mają atrybuty komenda **ls –l, ls –s, ls -F**	**- - -**	**r- x**
utworzenie nowego pliku	**- - -**	**- wx**
zmiana nazwy pliku	**- - -**	**- wx**
usunięcie pliku	**- - -**	**- wx**
czytanie pliku	**r - -**	**- - x**
zapis do pliku istniejącego	**- w -**	**- - x**
wykonywanie pliku binarnego	**- -x**	**- - x**
wykonywanie skryptu powłoki	**r - x**	**- - x**

Polecenia umożliwiające zmianę praw dostępu

Do zmiany uprawnień użytkowników w stosunku do pojedynczego pliku służy polecenie **chmod**. Wymaga ono określenia, czyje uprawnienia należy zmienić, na jakie i w stosunku do którego pliku. Prawa dostępu mogą być podane numerycznie (w formacie ósemkowym) albo symbolicznie. Podając prawa dostępu numerycznie korzystamy z polecenia:

chmod numeryczny_kod_uprawnien nazwa pliku

Prawo		binarnie	dziesiętnie
czytania	r - -	1 0 0	4
pisania	-w -	0 1 0	2
wykonywania	- - x	0 0 1	1

Punkty przysługujące poszczególnym kategoriom użytkowników należy złożyć razem. Prawa dostępu w postaci **rw-r--r--** interpretujemy następująco:

prawo dostępu właściciela	rw-	4+2+0=6
prawa dostępu grupy	r--	4+0+0=4
prawa dostępu innych	r--	4+0+0=4

Cały zestaw praw należy przedstawić jako liczbę trzycyfrową 644 – jest to numeryczny kod uprawnień i podać jako pierwszy argument komendy **chmod**, czyli **chmod 644 nazwa_pliku**.

Prawa dostępu można również ustawiać poleceniem:

chmod kto_ uprawnienia nazwa pliku

Blok **kto_uprawnienia** składa się z trzech elementów:

- Klasy praw dostępu, która określa komu nadawane są prawa i może być jednym, bądź kilkoma, spośród symboli:
  - **a –** wszyscy użytkownicy,
  - **u –** właściciel pliku,
  - **g –** grupa pliku,
  - **o –** inni użytkownicy.

Pominięcie symbolu kategorii nadaje wszystkim (właścicielowi, grupie, pozostałym) takie same prawa.

- Operatora, który jest jednym z następujących znaków:

– oznacza odebranie prawa,
+ oznacza dodanie prawa,
= ustala nowe uprawnienia niezależnie od stanu poprzedniego.

- Typu praw dostępu, który jest jednym, bądź kilkoma, spośród symboli: **r, w, x**.

Przykład

chmod u-w plik1 – zabiera **właścicielowi** prawo **pisania** do pliku **plik1**,

chmod a=rw plik1 – nadaje **wszystkim** prawo **rw** do pliku **plik1**,

chmod u+w, g-x plik1 – dodaje właścicielowi prawo **pisania** do pliku **plik1,** grupie odbiera prawo wykonania tego pliku,

chmod u=w,og+r-x plik1 – nadaje **właścicielowi** prawo **pisania** do pliku **plik1**, a członkom grupy, która jest grupą pliku oraz pozostałym użytkownikom systemu dodaje prawo czytania i odbiera prawo wykonania,

chmod 644 plik1 – nadaje prawo **rw** dla właściciela i prawo **r** dla wszystkich innych i jest równoznaczna komendzie **chmod u=rw,og=r plik1**.

Można zestawiać ze sobą kilka ciągów znaków reprezentujących nowe prawa, oddzielając je przecinkami.

Mimo że inni użytkownicy mogą mieć dostęp do pliku, jedynie jego właściciel może zmienić jego prawa dostępu. Jeśli chcemy dać innym użytkownikom kontrolę nad prawami dostępu do jednego z plików, poleceniem **chown** można zmienić właściciela pliku na innego użytkownika.

Polecenie **chown** przekazuje kontrolę nad plikiem innemu użytkownikowi. Jako pierwszy argument przyjmuje nazwę innego użytkownika. Za nazwą tą można umieścić listę plików, nad którymi przekazuje się kontrolę. Polecenie ma następującą postać:

chown opcje właściciel pliku nazwa pliku

Poleceniem **chown ania plik1 –** użytkownik daje kontrolę nad plikiem **plik1** Ani.

Można również zmienić grupę pliku za pomocą polecenia **chgrp**. Polecenie to przyjmuje za pierwszy argument nazwę nowej grupy dla pliku lub plików:

chgrp opcje grupa nazwy plików

Poleceniem, postaci **chgrp grupa_1 plik1 plik2**, użytkownik zmienia grupę plików **plik1** i **plik2** na **grupa_1**.

Rozszerzone prawa dostępu do pliku

Prawo **x** na poziomie użytkownika lub grupy może być zastąpione przez **s** (dla użytkownika nosi ono nazwę **SUID**, natomiast dla grupy **SGID**). Znaczenie tych praw zostanie omówione na laboratorium dotyczącym procesów i ich atrybutów.

Prawo **x** może być zastąpione także literą **t** (ang. save **t**ext mode). Prawo to (sticky bit **–** lepki bit) w przypadku katalogów oznacza, że pliki z tego katalogu może wymazać użytkownik posiadający jedynie prawo pisania do pliku lub właściciel pliku (stosowane np. dla systemowego katalogu tmp). W przypadku plików powoduje to, utrzymanie w pamięci binariów programu nawet po zakończeniu procesu, który z nich korzystał.

Ćwiczenia 5

1. W katalogu /home/fa001 znajdują się pliki **plik1** oraz **plik2**. Ustawione są do nich prawa dostępu **rw-r-----**. Chcesz skopiować plik **plik1** do katalogu **/home/fa002** oraz przenieść **plik2** do katalogu **/home/fa002**. Opisy plików **fa001** i **fa002** z katalogu **/home** mają postać:
   drwxr----- 10 fa001 fitec-00 1024 May 7 9:15 fa001
   drwxr----- 10 fa002 fitec-00 1024 May 7 9:15 fa002

Jakie są niezbędne prawa dostępu umożliwiające wykonanie tych poleceń, jeśli jesteś właścicielem pliku **fa001**? Podaj wszystkie polecenia konieczne do wykonania tych operacji z zaznaczeniem, przez którego użytkownika mają one być wykonane.

2. Jakie prawa zostaną nadane plikowi **p1** poleceniami:
   chmod 574 p1 **jeśli posiadał prawa rwxr-----**
   chmod u=rw,g=r,o-r p174 p1 **jeśli posiadał prawa rw--w-rw-**
3. Utwórz w swoim domowym katalogu dwa pliki o nazwach: pisz oraz czytaj. Plik pisz udostępnij kolegom z twojej grupy tak, aby mieli możliwość jego edycji, natomiast czytaj jedynie w trybie oglądania zawartości bądź możliwości skopiowania. Pamiętaj o prawach do katalogu.
4. Sprawdź, czy możesz usunąć plik, do którego nie masz żadnych praw dostępu.
5. Jakie komendy zmieniają liczbę dowiązań do pliku? Sprawdź to.
6. Wypełnij poniższą tabelkę zaznaczając symbolami „+" prawa dostępu konieczne do wykonania poleceń 1 **–** 13.

Założenia ogólne:

- istnieją katalogi **kat1**, **kat2**,

Założenia do punktów 1 – 11:

- istnieje plik **pl1** w katalogu **kat1**,
- istnieje plik **pl2** w katalogu **kat2**.

Założenia do punktów 12 – 13:

- istnieje plik **pl1** w katalogu **kat1**,
- nie istnieje plik **pl2** w katalogu **kat2**

nr	**1**			**2**			**3**			**4**			**5**			**6**			**7**		
pr.	r	w	x	r	w	x	r	w	x	r	w	x	r	w	x	r	w	x	r	w	x
kat1																					
pl1																					
kat2																					
pl2																					

nr	**8**			**9**			**10**			**11**			**12**			**13**		
pr.	r	w	x	r	w	x	r	w	x	r	w	x	r	w	x	r	w	x
kat1																		
pl1																		
kat2																		
pl2																		

1. **cd kat1**
2. **ls kat1**
3. **ls –s kat1**
4. **ls –F kat1**
5. **ls –i kat1**
6. **vi kat1/pl1 (pełna edycja pliku edytorem – możliwość odczytu i zapisu modyfikacji)**
7. **cat kat1/pl1**
8. **cat > kat1/pl1**
9. **rm kat1/pl1**
10. **cp kat1/pl1 kat2/pl2**
11. **mv kat1/pl1 kat2/pl2**
12. **ln kat1/pl1 kat2/pl2**
13. **ln –s kat1/pl1 kat2/pl2**

*Treści oznaczone kursywą pochodzą z różnych źródeł internetowych.