

WPROWADZENIE DO UNIX'A / LINUX'A (4 – PLIKI I KATALOGI)

1. Logowanie się

W tym celu korzystamy z programu PuTTY. W polu Host Name wpisujemy: `tempus.metal.agh.edu.pl`

2. Struktura katalogów

Aby poznać umiejscowienie naszego katalogu bieżącego (sprawdzić gdzie się aktualnie znajdujesz) w strukturze katalogów całego systemu, wprowadź komendę `pwd` (ang. print working directory). Wynik będzie podobny do:

```
/home/prac/regulski/infor1/katalog01
```

Aby wyświetlić zawartość katalogu, wypróbuj:

`ls` – (ang. list) wyświetla zawartość katalogu.

`ls -a` – wyświetlanie całej zawartości katalogu bieżącego

`ls /` – wyświetlanie zawartości katalogu głównego

`ls ..` – wyświetlanie zawartości katalogu nadrzędnego

`ls /nazwakatalogu` – wyświetlanie zawartości dowolnego katalogu

`ls -l` – wyświetlanie szczegółowe zawartości katalogu

Do przemieszczania się po drzewie katalogów służy polecenie `cd` (change directory). Wypróbuj:

```
cd <nazwa katalogu>
```

```
cd ..
```

```
cd <katalog>/<podkatalog>
```

```
cd /
```

```
pwd
```

Struktura katalogów `Unixa` jest stała i nienaruszalna. Może ona wyglądać jak na rysunku, lecz w zależności od wersji `Unixa` może być nieznacznie zmodyfikowana. Każdy katalog ma określoną funkcję i użytkownicy korzystają z niego w zakresie przydzielonych im uprawnień.

Struktura katalogów w Unixie różni się od struktury Windows'a m.in. tym, że wszystkie urządzenia takie jak FDD, CD-ROM, HDD itd. są reprezentowane przez inne katalogi.

`/bin` - programy wykonywalne

`/dev` - pliki urządzeń komputera, portów, terminali

`/etc` - pliki i narzędzia administracyjne systemu, skrypty startujące i zatrzymujące system, podsystemy autoryzacji i ustawień jądra systemu

`/home` - Katalog `/home` jest katalogiem najważniejszym z punktu widzenia szarego użytkownika systemu. W tym katalogu znajdują się najczęściej katalogi domowe użytkowników.

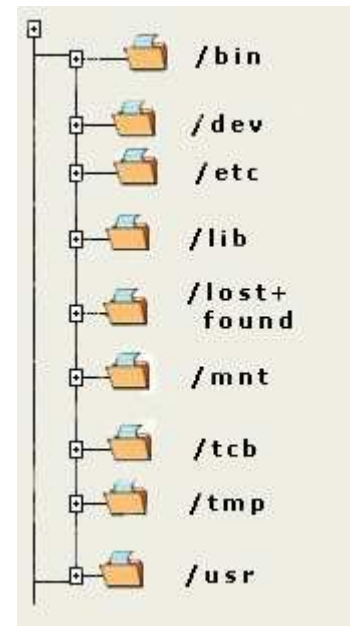
`/lib` - biblioteki programistyczne używane przez programy i narzędzia

`/mnt` - katalog do montowania FDD, CD-ROM, HDD itd.

`/tc` - informacje o zabezpieczeniach systemu, hasła i logi użytkowników

`/tmp` - katalog plików tymczasowych

`/usr` - Katalog `/usr` jest prawdopodobnie najbogatszym w różne dobra katalogiem w systemie. Znajdują się tam różne pliki, od źródeł poczynając na dokumentacjach skończywszy.



3. Polecenie `wget`

Polecenie `wget` pozwala pobierać z podanego adresu URL pliki, katalogi i całą strukturę stron WWW.

```
wget [opcje] [lista adresów URL]
```

W swoim **katalogu domowym** pobierz teraz materiały ze strony:

```
wget -r http://tempus.metal.agh.edu.pl/~regulski/dydaktyka/informatyka/pliki/index.htm
```

Znajdź w nowych katalogach pliki:

```
alfa.xls    index.html    schemat.txt    tem011.dat    tnaroz.dat
faktura.pdf  pracownicy.sql    tekst.txt    tld12012.dat
```

4. Wyświetlanie zawartości pliku poleceniem `cat`

```
cat nazwa_pliku
```

Polecenie `cat` służy tak naprawdę do łączenia plików.

np: `cat plik1 plik2 > suma_plików`

spowoduje połączenie plików `plik1`, `plik2`, w podanej kolejności i zapisanie wyników w nowo utworzonym pliku o nazwie `suma_plików`, a jeżeli istniał, usunięcie jego zawartości przed zapisem.

Gdy zamiast `>` użyjemy `>>` spowoduje to jedynie dołączenie plików bez uprzedniego usuwania zawartości.

Zadanie 1.

Odszukaj w swoim katalogu domowym katalog ze ściągniętymi ze strony plikami, wejdź do niego, a następnie wypróbuj działanie poleceń:

```
cat schemat.txt
cat pracownicy.sql
cat schemat.txt pracownicy.sql > plik.cat
cat plik.cat
cat tld12012.dat
cat plik.cat >> tld12012.dat
cat tld12012.dat
```

Wypróbuj działanie polecenia w formie:

```
cat > nowy.plik [Enter]
"Napisz dowolną wiadomość"
CTRL+D
cat nowy.plik
```

5. Wyświetlanie zawartości pliku poleceniem `more`

```
more nazwa_pliku
```

Pozwala na przeglądanie zawartości pliku "strona po stronie". Wznowienie wyświetlania następuje po wciśnięciu klawisza `d` (przesunięcie o pół ekranu) lub `spacji` (przesunięcie o cały ekran) lub `enter` (przesunięcie o cały wiersz).

Użytkownik może zadeklarować liczbę wierszy w porcji wyświetlanego tekstu.

np: `more -20 nazwa_pliku` spowoduje, że jednorazowo wyświetlanych będzie 20 wierszy.

Polecenie `more` posiada także pewną ilość opcji. Oto niektóre z nich:

`-c` powoduje każdorazowe czyszczenie ekranu przed wyświetleniem kolejnej porcji tekstu,

`-f` powoduje, że długie wiersze tekstu (zbyt długie, aby mogły zmieścić się na ekranie) nie są przelamywane, a na ekranie pojawia się tylko ich fragment - tyle znaków ile zmieści się na monitorze.

Program `more` pozwala na przeglądanie od wiersza o określonym numerze, np od wiersza setnego:

```
more +100 nazwa_pliku
```

Program `more` pozwala również na wyświetlanie od wiersza zawierającego tekst pasujący do podanego wcześniej wzorca. Postać polecenia `more` jest wtedy następująca:

```
more +/wzorzec nazwa_pliku
```

Wypróbuj: `more tem011.dat`

6. Kopiowanie, przesuwanie, usuwanie plików

Kopiowanie:

```
cp plik_wejściowy plik_wyjściowy
```

np.: `cp plik.cat plik_kopia.moja.kopia`

Przesuwanie:

```
mv plik_wejściowy plik_wyjściowy
```

np.:

```
mv plik_kopia.moja.kopia /home/tmp/im2009/[login]/skopiowany_przesuniety.plik
```

Zmiana nazwy:

Wejść do katalogu domowego, sprawdź, czy jest tam plik

```
skopiowany_przesuniety.plik
```

zmień jego nazwę za pomocą polecenia:

```
mv skopiowany_przesuniety.plik zmieniona.nazwa
```

Usuwanie:

```
rm [-irf] nazwa_pliku
```

Występują tutaj trzy opcje:

`-f` zmusza polecenie do usunięcia plików, których jesteśmy właścicielem a te, do których nie mamy ustawionego prawa zapisu usuwa bez wprowadzania komunikatu decyzyjnego (wymagającego dodatkowego potwierdzenia)

`-i` powoduje wprowadzenie komunikatu decyzyjnego przed każdym usunięciem pliku

`-r` powoduje usuwanie plików i podkatalogów, łącznie z wyspecyfikowanym katalogiem

Wypróbuj:

```
rm zmieniona.nazwa
```

7. Co to są metaznaki (wildcards)?

Są to specjalne znaki, dzięki którym można tworzyć wzorce ciągów znaków (wzorniki, wyrażenia regularne) używane m.in. do operacji na plikach.

* — oznacza dowolny ciąg znaków

? — oznacza pojedynczy, dowolny znak

Wejść do katalogu z pobranymi plikami.

np. możemy wyświetlić wszystkie pliki tekstowe w katalogu używając polecenia

```
ls -F *.txt
```

znaki [] oraz -

W nawiasach możemy umieścić znaki lub przedział znaków, które mogą znaleźć się na danej pozycji.

Szczegóły dotyczące plików z rozszerzeniem `.dat` mające w nazwie na drugiej pozycji literę `e` lub `l` lub `n` wskażemy poleceniem:

```
ls -l ?[e-l]*.dat
```

W katalogu bieżącym znajdują się pliki:

```
alfa.xls, faktura.pdf, pracownicy.sql, schemat.txt, tekst.txt,  
tem011.dat, tld12012.dat, tnaroz.dat
```

Jeżeli chcemy usunąć grupę plików, np. `alfa.xls`, `faktura.pdf`, to nie musimy usuwać ich pojedynczo, ale możemy to zrobić za jednym razem:

```
rm [a-f]*.*
```

Polecenie:

```
ls t?[dm]*.*
```

wyświetli pliki o nazwach `tem011.dat` `tld12012.dat`.

Polecenie:

```
ls t?[d-m]*.*
```

wyświetli pliki o nazwach `tekst.txt`, `tem011.dat`, `tld12012.dat`.

Inne przykłady:

```
rm plik — Kasuje plik.
```

```
rm * — Kasuje wszystkie pliki w danym katalogu.
```

```
rm * -i — Kasuje wszystkie pliki w danym katalogu z potwierdzeniem.
```

```
rm * -f — Kasuje wszystkie pliki w danym katalogu bez pytania.
```

```
rm -r — Kasuje wszystkie pliki, także te w podkatalogach
```

```
rm -rf /home/tmp/im2009 — Kasuje wszystkie pliki i katalogi w katalogu  
/home/tmp/im2009
```

8. Jak odszukać plik?

Aby odszukać interesujący nas plik możemy posłużyć się poleceniem **find**. Polecenie to posiada bardzo wiele kluczy (wyrażeń), według których możemy odnaleźć plik lub grupę plików.

```
find katalog kryterium_wyszukiwania
```

Katalog określa wierzchołek drzewa do przeszukania (przeszukanie obejmuje jego wszystkie podkatalogi). Bieżący katalog może być oznaczony jako kropka ".", nadrzędny jako dwie kropki "..".

Wpisanie "/" spowoduje przeszukanie wszystkich katalogów.

kryterium_wyszukiwania jest zbiorem warunków ustalającym cechy charakterystyczne szukanego pliku, może to być nazwa pliku. Przykłady:

```
-name nazwa_pliku          pliki o podanej nazwie
-iname nazwa_pliku        j.w. bez zwracania uwagi na wielkość liter
-user użytkownik          pliki, których właścicielem jest podany użytkownik
-exec polecenie {} \;     dla każdego odnalezionego pliku wykonaj polecenie
```

Np. jeżeli znana jest nazwa pliku polecenie

```
find .. -name lista1
```

spowoduje rozpoczęcie przeszukiwania od katalogu nadrzędnego w stosunku do katalogu bieżącego, w celu znalezienia pliku **lista1** (opcja **-name**). Po znalezieniu tego pliku na ekranie będzie wypisana pełna nazwa ścieżki. Polecenie

```
find ./ -name l"*text*" mtime -5 -atime +1 -print
```

spowoduje wypisanie listy nazw ścieżek wszystkich plików zawierających w nazwie ciąg znaków **text**, zmodyfikowanych nie wcześniej niż 6 dni temu, a z których korzystano co najmniej 2 dni temu. Jeżeli chcemy podać dokładną liczbę dni, które upłynęły od ostatniej modyfikacji (**mtime**) lub korzystania (**atime**), podajemy po prostu tę liczbę. Znak **+** oznacza okres dłuższy od podanego, a **-** krótszy.

Polecenia **find / -nouser -exec rm {} \;** - w całym drzewie katalogów znajdzie i usunie wszystkie pliki, których właściciel nie istnieje.

Zadanie: Znajdź w swoim katalogu plik **style.css**

grep [opcja] wyrażenie nazwa_pliku – służy do wyszukiwania w plikach poszczególnych wersów zawierających dane wyrażenie.

Zadanie: **grep -in nazwisk *** - opcja **-i** 'znieczula' na wielkość liter, opcja **-n** nakazuje wypisać numer linii w pliku

9. Inne operacje na plikach

`head` — wyświetl tylko początek pliku. Można podać liczbę wierszy

`tail` — wyświetl tylko ostatnie linie pliku

`wc` — zlicza słowa w pliku tekstowym, linie tekstu i/lub znaki w pliku

`tr` — pozwala z linii komend edytować plik zamieniając w nim znaki. np.:

`tr '[abcde]' [12345] < plik .txt` — Wstaw za 'a' - '1', za 'b' - '2' itd aż do pary 'e' - '5', zgodnie z kolejnością występowania tych znaków w nawiasach kwadratowych.

`tr '[: lower :]' '[: upper :]' < plik .txt` — zamień wszystkie małe litery na duże

`tr '\t' ' ' < plik .txt` — zamień wszystkie znaki tabulacji na spacje. Inny ważny znak specjalny: `\n` znak końca linii

`sort` — sortuje linie w pliku alfabetycznie (numerycznie)

`uniq` — usuwa duplikaty wierszy w pliku

`cut` — wycina określone kolumny, które można definiować jako listę znaków w wierszu (np. `-c 1,3,5,7,9-14`) bądź podając numer pola (poprzez opcję `-f`). W tym drugim przypadku trzeba zdefiniować także separator (opcja `cut -d`), czyli znak, który oddziela jedno pole od drugiego (np. `-f 3 -d ':'` - w tym przypadku separatorem jest znak ':').

`cut -c 30 -54 1 ixA .pdb` — Wytnij kolumny 30-54 z pliku pdb.

`paste` — składanie kilku kolumn w jeden plik

`paste -d ':' plik1 .txt plik2 .txt` — Scal kolumnami pliki plik1.txt i plik2.txt, używając znaku ':' jako separatora.

`fold` — zawijanie wierszy do określonej liczby kolumn

`gzip (gunzip)` — skompresuj plik

`tar` — skompresuj katalog w jeden plik

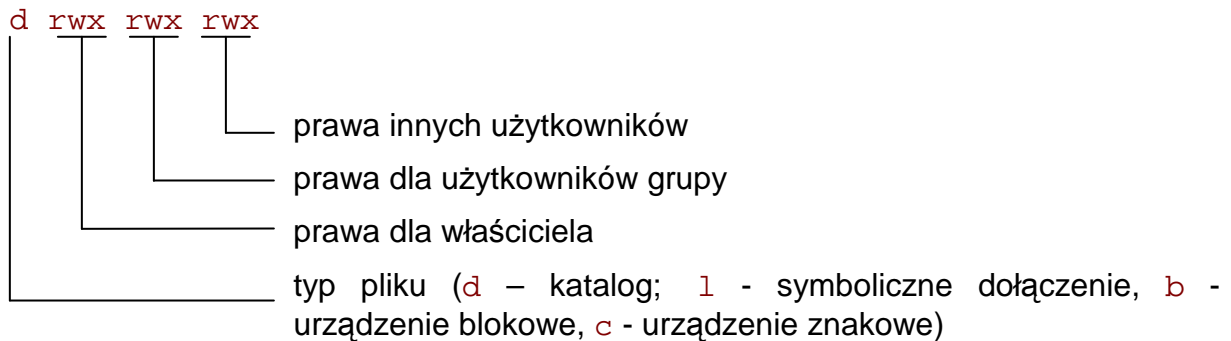
`touch` — modyfikowacja znacznika czasu pliku, częściej jednak używa się jej do tworzenia pustych plików. `touch pusty_plik`

`diff plik1 plik2` — porównanie dwóch plików

10. Prawa dostępu

Każdy plik i kartoteka w systemie UNIX posiada atrybuty, określające prawa dostępu do takiego obiektu. Prawa dostępu są definiowane w momencie tworzenia obiektu - według maski ustawionej przez administratora systemu.

Właścicielem obiektu jest ten, kto go stworzył. Właściciel obiektu może w każdej chwili zmienić prawa dostępu do obiektu - zarówno dla siebie jak i dla innych użytkowników.



- `r` — prawo czytania plików (read)
- `w` — prawo do modyfikacji plików (write)
- `x` — prawo wykonywania (execute)
- `-` — brak zezwolenia

Przez **prawo odczytu** rozumieć należy możliwość odczytania zawartości pliku (co w przypadku katalogu oznacza możliwość odczytania listy plików znajdujących się w tym katalogu).

Prawo zapisu pozwala na dokonanie modyfikacji pliku (w tym dopisanie, zmiana lub usunięcie zawartości pliku. Przy czym usunięcie zawartości pliku nie jest tożsame z fizycznym usunięciem pliku). Dla katalogów to prawo oznacza możliwość tworzenia nowych i usuwania istniejących plików w katalogu.

Prawo wykonywania pliku pozwala na uruchomienie pliku wykonalnego, a w przypadku katalogu oznacza prawo dostępu do plików wewnątrz katalogu.

Czynność	Plik	Katalog nadrzędny
przeglądanie zawartości katalogu	---	r--
utworzenie, skasowanie, zmiana nazwy pliku	---	-w-
zmiana danego katalogu na bieżący	---	--x
czytanie pliku	r--	--x
zapis do pliku	-w-	--x
wykonanie pliku	--x	--x

Prawa dostępu są definiowane na trzech poziomach:

- dla właściciela obiektu (**u**)
- dla grupy, do której należy właściciel obiektu (**g**)
- dla pozostałych użytkowników (**o**)

Jak wydasz polecenie `ls -l` to wyświetli się coś takiego:

```
-rw-r--r-- 1 root root 32251 Feb 15 14:32 DOS2Linux
-rw-r--r-- 1 root root 38338 Feb 20 13:15 DOS2Linux.pl.shtml
-rw-r--r-- 1 root root 18144 Feb 19 18:34 Locales-HOWTO.shtml
-rw-r--r-- 1 root root 13980 Feb 19 18:30 howto.dat
```

Do zmiany praw dostępu służy komenda:

```
chmod komu jak prawa wzornik
```

```
chmod ugoa+--rwx <plik>
```

komu - dowolna kombinacja liter **u**, **g** i **o** oznaczających kolejno właściciela pliku, członków grupy do której należy właściciel pliku (z wyłączeniem właściciela obiektu) oraz pozostałych użytkowników. Dodatkowo można użyć litery **a**, gdzie **a** oznacza wszystkich użytkowników (**a** = **ugo**)

jak - jeden ze znaków "+", "-", "=" oznaczających:

+ dodaj prawo

- usuń prawa

= usuń wszystkie wcześniej ustanowione prawa dostępu do obiektu, po usunięciu nadaj wyłącznie te prawa, które są wymienione w komendzie

prawa - dowolna kombinacja praw **r**, **w**, **x**.

wzornik - obiekty, których nazwy są zgodne z wzornikiem będą tymi obiektami, dla których zmieniamy prawa dostępu (np. nazwa pliku).

Przykłady:

```
chmod u+x <plik> - nadaje właścicielowi pliku prawo do wykonywania go.
```

```
chmod ug+rx <plik> - nadaje właścicielowi i grupie, do której należy plik prawa do odczytu i wykonywania
```

```
chmod go-rwx <plik> - zdjęcie wszystkich praw dla grupy i innych użytkowników.
```

Zadanie: zmień prawa dostępu do pliku (stworzonego z użyciem polecenia `cat`) `nowy.plik` na pełne prawa dla wszystkich.

SPOSÓB 2 (numeryczny)

Podstawmy sobie następujące wartości:

```
-  rwx  rwx  rwx
   421  421  421
```

Żeby ustawić jakieś prawa dostępu należy dodać do siebie te liczby, które stoją przy prawach, które chcemy ustawić, ale dla każdej trójki osobno. Trochę przykładów dla lepszego zrozumienia:

```
-  r-x --- --- nazwa_pliku
   4 1
```

To będzie 4+0+1, 0+0+0, 0+0+0 czyli 500. `chmod 500 nazwa_pliku` zmieni prawa dostępu na odczyt i wykonywanie dla właściciela i zabierze wszystkie prawa dla grupy i innych użytkowników.

```
-  rwx r-x r-x nazwa_pliku
   421 4-1 4-1
```

To będzie 4+2+1, 4+0+1, 4+0+1 czyli 755 - to jest chyba najczęstsze ustawienie.

Zadanie: zmień prawa dostępu do pliku `nowy.plik` (sposobem numerycznym) na brak praw dla grupy i pozostałych użytkowników.

`chown`, `chgrp` – polecenia zmieniają właściciela pliku oraz grupę przynależności pliku

11. Operacje na katalogach

`mkdir` — tworzenie katalogu

```
mkdir ścieżka_dostępu/nazwakatalogu
```

Jeśli podamy tylko nazwę katalogu zostanie on utworzony w katalogu bieżącym.

`rmdir nazwa_katalogu` — usuwamy pusty katalog o podanej nazwie

`rmdir --ignore-fail-on-non-empty [Katalog1]` — usuwa katalogi bez komunikatu o błędzie, jeśli niepusty

`rmdir --ignore-fail-on-non-empty -p [Katalog1/Katalog2/Katalog3]` — usunięcie katalogu wraz z katalogami nadrzędnymi, aż do katalogu w którym znajdują się pliki.

`mv` — zmiana nazwy lub przeniesienie katalogu

```
mv [źródło] [cel]
```

`mv /home/user /tmp` — katalog `/home/user` zostanie przeniesiony do `/tmp`.

`mv plik1 plik2` - 'plik1' zmieni nazwę na 'plik2'

`rm -rv nazwa_katalogu` — usuwa niepuste katalogi (rekurencyjnie) wyświetla przy tym nazwy usuniętych plików

12. Midnight Commander

`Midnight Commander` jest nakładką systemową podobną do `Norton Commandera` znanego z systemu `DOS` lub `Windows Commander` z `Windows`. Umożliwia przeglądanie katalogów, zarządzanie plikami (kopiowanie, usuwanie, przenoszenie i zmiana nazwy, zmiana atrybutów plików), edycję tekstów a także uruchamianie innych programów. Program `mc` uruchamiamy wpisując `mc` w linii poleceń.

Po uruchomieniu programu (polecenie: `mc`) pojawia się główne okno MC, podzielone na dwa panele. W każdym z paneli możemy "zrobić" to samo. O tym, w którym panelu obecnie pracujemy informuje nas "belka" (podświetlenie). Pomiędzy panelami przemieszczamy się używając przycisku `Tab` (tabulator). Aby wykonać jakąś operację na pliku lub katalogu, należy ustawić belkę na danym pliku i użyć właściwego klawisza funkcyjnego (`F1` - `F10`). Każdy z tych klawiszy posiada przypisaną na stałe akcję - są to:

`F1` - wyświetlenie pomocy

`F2` - dostęp do menu użytkownika (m.in. opcja kompresowania katalogów i plików)

`F3` - podgląd zawartości pliku

`F4` - edycja pliku (`SHIFT+F4` - utworzenie nowego pliku tekstowego)

`F5` - kopiowanie plików

`F6` - przenoszenie / zmiana nazwy plików

`F7` - tworzenie katalogu

`F8` - usuwanie plików

`F9` - rozwinięcie górnego menu

`F10` - zakończenie programu

Znaczenie (działanie) każdego z klawiszy funkcyjnych zapisane jest (zawsze) u dołu terminala.

`CTRL+o` pozwala na chwilę ukryć panele

Zadania:

W swoim katalogu domowym utwórz katalog o nazwie `tydzien`

w katalogu `tydzien` utwórz następującą strukturę katalogów:

```
~/-- tydzien
  |-- poniedzialek
  |-- wtorek
  |-- sroda
  |-- czwartek
  |           |-- ranek
  |           |-- wieczor
  |           |           |-- siedemnasta
  |           |           |-- dziewietnasta
  |           |
  |           |-- noc
  |
  |-- piatek
  |-- sobota
  |           |-- ranek
  |           |-- wieczor
  |           |-- noc
  |
  |-- niedziela
```

W katalogu `~/tydzień/czwartek/wieczor/siedemnasta` stwórz plik `odrabianie.zadania`

W katalogu `~/tydzień/czwartek/wieczor/dziewietnasta` stwórz plik `dobranocka`

W katalogu `~/tydzień/sobota/wieczor/` stwórz plik `impreza` i nadaj mu prawa dostępu dla wszystkich (`rwX`)

W katalogu `~/tydzień/niedziela` stwórz plik `kac` o treści 'Nigdy więcej'.

Usuń katalog `niedziela` i katalog `poniedzialek`.

Skompresuj cały katalog `tydzien` (i tylko ten katalog) i wyślij mailem w załączniku na adres `regulski@tempus.metal.agh.edu.pl`