

Karolina Spurek

Sprawozdania do zadań projektowych**Zadanie 1 - Niesforne dane**

cat dane.txt (wyświetla plik).

dos2unix plik.txt (przekonwertowanie pliku na format Unix)

paste -d "\t" - - - <dane.txt (sortuje dane)

paste -d "\t" - - - <dane.txt > dane123123.txt (do pliku txt)

sed -i 'li x\ty\tz' dane123123.txt

{dodaje nagłówki x y z; sed (stream editor – edycja pliku) ; -i (zmiany dokonywane bezpośrednio bez zapisu do nowego pliku); li x\ty\tz (1 nr pierwszej linii; i insert; x\ty\tz wstawiony tekst (\ tabulacja))}

Zadanie 2 - Dodawanie poprawek

diff -u lista.txt lista-pop.txt > lista.patch (porównuje oba pliki i zapisuje różnice w pliku patch)

dos2unix lista.txt (przekonwertowanie pliku na format Unix)

patch lista.txt < lista.patch (aplikowanie łatki)

md5sum lista-pop.txt lista.txt (md5sum sprawdza integralność między nowo poprawionym plikiem lista.txt a lista-pop.txt. Ciągi liczb i cyfr zgadzają się w obu plikach co oznacza że zadanie zostało poprawnie wykonane).

Zadanie 3 - Z CSV do SQL i z powrotem

cat steps-2sql.csv (wklejanie pliku do msys)

tail -n +2 steps-2sql.csv | awk -F";" '{printf "INSERT INTO stepsData (time, intensity, steps) VALUES (%s, %s, %s);\n", \$1, \$2, \$3}> steps-2sql-wynik.sql

{tail -n +2 - pomija nagłówki CSV.

awk -F";" - ustawia separator pól na ;.

printf - formatuje dane jako zapytania SQL. }

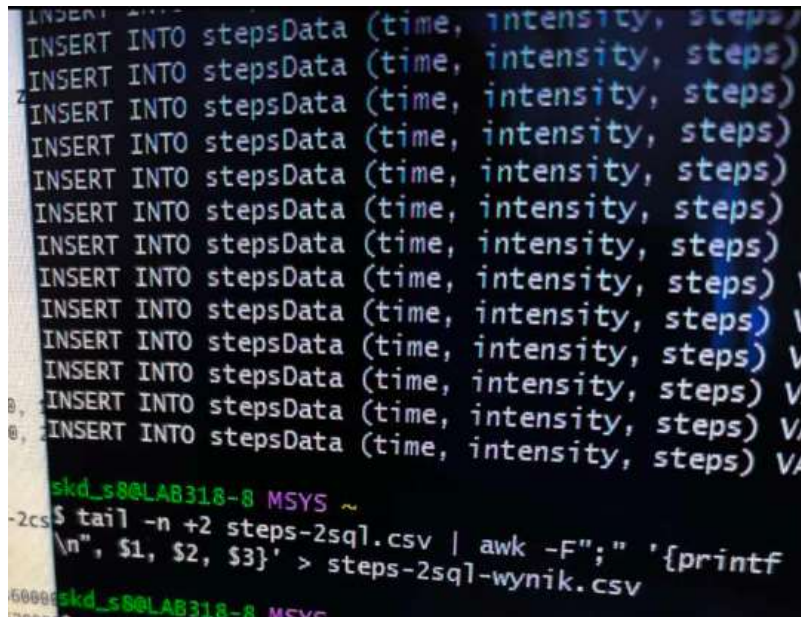
```
echo "dateTime; steps; synced"
```

```
grep "INSERT INTO" steps-2csv.sql | sed -E 's/.*VALUES \([([0-9]+), ([0-9]+), ([0-9]+)\);  
/1; \2; \3/' | awk -F";" '{printf "%d;%s; %s\n", $1/1000, $2, $3}' > steps-2csv-wynik.csv
```

{grep - wybiera linie z INSERT INTO.

sed - wyciąga liczby z VALUES (...).

awk - dzieli przez 1000 (usuwa milisekundy) i zapisuje jako CSV. }



Zadanie 4 - Marudny tłumacz

```
echo "((" >pl-7.2.4.json5 (tworzymy nowy plik)
```

```
grep -E '^\\s*"^[^"]+:\\s*"^[^"]+" en-7.2.json5 | sed -E 's/^(\\s*)"([^[^"]+):\\s*"([^[^"]+)",?/\\1W  
"\\2": "\\3", \\n\\1"2": "\\3",/" >> pl-7.2.4.json5
```

{ grep wyszukuje tylko linie, które wyglądają jak wpisy w formacie JSON.

-E - używamy rozszerzonych wyrażeń regularnych.

^\\s*" - linia zaczyna się od dowolnej liczby spacji, potem cudzysłów.

[^"]+ - dowolny ciąg nie będący cudzysłowem.

:\\s*" - potem dwukropek i wartość w cudzysłowie.

sed - przekształca każdą linię JSON w komentarz + oryginalny wpis.

^(\\s*)" - zapamiętuje w zmiennej \\1 wcięcie.

"([^[^"]+)" - zapamiętuje klucz, wartość.

,? - dopuszcza przecinek na końcu.

\1// "\2": "j3", - dodaje komentarz,

\n\1"\2": "\3", - oryginalna linia po komentarzu, z zachowaniem wcięcia. }

echo "}" >> pl-7.2.4.json5 (dopisuje przetworzone linie do wcześniej utworzonego pliku)

A screenshot of a terminal window with a dark background. The terminal shows a JSON5 file content with various workflow and session management items. The text is as follows:

```
"workflow-item.delete.notification.success.content": "This workflow item was successfully deleted",
// "workflow-item.delete.notification.error.title": "Something went wrong",
"workflow-item.delete.notification.error.title": "Something went wrong",
// "workflow-item.delete.notification.error.content": "The workflow item could not be deleted",

"workflow-item.delete.notification.error.content": "The workflow item could not be deleted",
// "workflow-item.delete.title": "Delete workflow item",
"workflow-item.delete.title": "Delete workflow item",
// "workflow-item.delete.header": "Delete workflow item",
"workflow-item.delete.header": "Delete workflow item",
// "workflow-item.delete.button.cancel": "Cancel",
"workflow-item.delete.button.cancel": "Cancel",
// "workflow-item.delete.button.confirm": "Delete",
"workflow-item.delete.button.confirm": "Delete",
// "workflow-item.send-back.notification.success.title": "Sent back to submitter",
"workflow-item.send-back.notification.success.title": "Sent back to submitter",
// "workflow-item.send-back.notification.success.content": "This workflow item was successfully sent back to the submitter",
"workflow-item.send-back.notification.success.content": "This workflow item was successfully sent back to the submitter",
// "workflow-item.send-back.notification.error.title": "Something went wrong",
"workflow-item.send-back.notification.error.title": "Something went wrong",
// "workflow-item.send-back.notification.error.content": "The workflow item could not be sent back to the submitter",
"workflow-item.send-back.notification.error.content": "The workflow item could not be sent back to the submitter",
// "workflow-item.send-back.title": "Send workflow item back to submitter",
"workflow-item.send-back.title": "Send workflow item back to submitter",
// "workflow-item.send-back.header": "Send workflow item back to submitter",
"workflow-item.send-back.header": "Send workflow item back to submitter",
// "workflow-item.send-back.button.cancel": "Cancel",
"workflow-item.send-back.button.cancel": "Cancel",
// "workflow-item.send-back.button.confirm": "Send back",
"workflow-item.send-back.button.confirm": "Send back",
// "workflow-item.view.breadcrumbs": "Workflow View",
"workflow-item.view.breadcrumbs": "Workflow View",
// "idle-modal.header": "Session will expire soon",
"idle-modal.header": "Session will expire soon",
// "idle-modal.info": "For security reasons, user sessions expire after {{ timeToExpire }} minutes of inactivity. Your session will expire soon. Would you like to extend it or log out?",
"idle-modal.info": "For security reasons, user sessions expire after {{ timeToExpire }} minutes of inactivity. Your session will expire soon. Would you like to extend it or log out?",
// "idle-modal.log-out": "Log out",
"idle-modal.log-out": "Log out",
// "idle-modal.extend-session": "Extend session",
"idle-modal.extend-session": "Extend session",
}
```

The terminal prompt at the bottom shows the user is 'skd_s8@LAB318-8' in the 'MSYS' environment, with a tilde '~' as the current directory. The cursor is on a new line after the prompt.

Zadanie 5 - Fotografik gamon

unzip *.zip (rozpakowuje plik o zip)

rm kopie-2.zip kopie-1.zip (usuwa oba pliki o nazwach kopie-1 i kopie-2)

unzip *.zip (rozpakowuje plik zip)

rm *.zip (usuwa wszystkie pliki zip)

for f in *.png; do

magick "\$f" "\${f%.png}.jpg";

done (wykonuje tą operację na każdym pliku .png i konwertuje go na plik jpg, \${f%.png}.jpg zamienia nazwę na taką z końcówką .jpg).














rm *.png (usuwa wszystkie pliki png)

mkdir -p zdjecia (tworzy katalog)

for f in *.jpg; do

magick "\$f" -resize x720 -units PixelsPerInch -density 96 "zdjecia/\$f";

done (zaznacza każdy plik jpg, ustawia wysokość każdego zdjęcia do 720 pikseli, zamienia jednostkę na "PixelsPerInch" i rozdzielczość na 96"dpi")

	adrien-olichon-3137064	27.05.2025 14:40	Plik JPG
	aleksandr-slobodianyik-...	20.01.2016 16:41	Plik JPG
	aleksandr-slobodianyik-...	25.01.2010 15:54	Plik JPG
	alexander-dummer-376...	27.05.2025 14:41	Plik JPG
	alexazabache-5117913	27.05.2025 14:41	Plik JPG
	alex-montes-892479-1...	27.05.2025 14:40	Plik JPG
	anni-roenkae-3435272	27.05.2025 14:41	Plik JPG
	anntarazevich-5620861	27.05.2025 14:41	Plik JPG
	archwall_dark_blue	27.05.2025 14:41	Plik JPG
	archwall_dark_orange	27.05.2025 14:41	Plik JPG
	archwall_dark_toxic	27.05.2025 14:41	Plik JPG
	archwall_dark_yellow	27.05.2025 14:41	Plik JPG
	arctic	27.05.2025 14:41	Plik JPG

Obraz

Identyfikator obrazu

Wymiary	1080 x 720
Szerokość	1080 pikseli
Wysokość	720 pikseli
Rozdzielczość w poziomie	96 dpi
Rozdzielczość w pionie	96 dpi

Zadanie 6 - Wszędzie te PDF-y

```
Kasia@DESKTOP-RHFFRLJ MINGW64 ~/zdjecia
$ mkdir -p podpisane

Kasia@DESKTOP-RHFFRLJ MINGW64 ~/zdjecia
$ for img in *.jpg; do
> filename=$(basename "$img")
> convert "$img" -gravity south -background white -splice 0x40 -font Arial -pointsize 24
  -fill black -annotate +0+10 "$filename" podpisane/"$filename"
> done
```

{mkdir –tworzy nowy katalog;

for img in ~/photos/*.jpg; do ... done - rozpoczyna pętlę w Bashu, która przejdzie po każdym pliku .jpg w katalogu zdjęcia;

filename=\$(basename "\$img") - wyciąga samą nazwę pliku i zapisuje ją w zmiennej filename;

convert "\$img" ... ~/photos/labeled/"\$filename" - wywołuje narzędzie ImageMagick convert, które edytuje jeden obraz i zapisuje wynik pod nową ścieżką;}

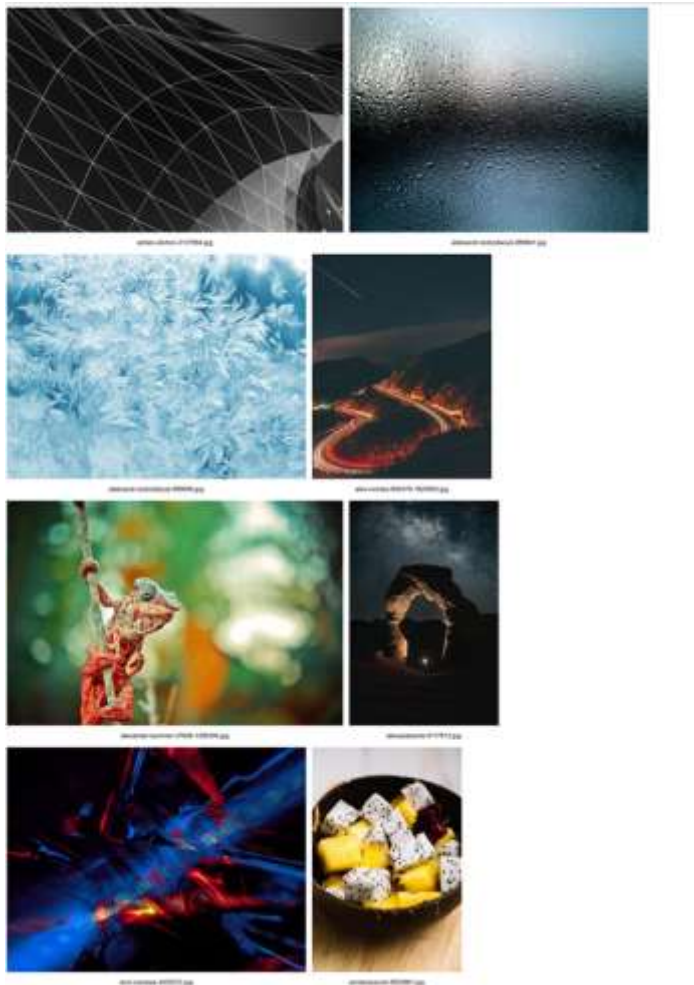
montage podpisane/*.jpg -tile 2x4 -geometry 360x220+10+10 -background white -gravity center portfolio_zdjec.pdf

{montage podpisane/*.jpg - uruchamia program montage z ImageMagick i przekazuje mu wszystkie obrazy JPG z katalogu podpisane;

-tile 2x4 - ustala, że na jednej stronie powstanie siatka z 2 kolumn i 4 wierszy, czyli razem maks. 8 zdjęć na stronie;

-geometry 360x220+10+10 - mówi, jak duże ma być „miejsce” dla każdego obrazka: 360 pikseli szerokości i 220 pikseli wysokości, +10+10 to odstęp między obrazami: 10 pikseli poziomo i 10 pikseli pionowo;

portfolio_zdjec.pdf - nazwa pliku wynikowego, do którego montage zapisze całość.}



Zadanie 7 - **Porządki w kopiach zapasowych**

cd kopie | (wchodzi w katalog kopie)

cd | (wraca do poprzedniego katalogu)

bash sortowanie.bat | (Otwiera i uruchamia skrypt batchowy zawarty w pliku .bat)

{Zawartość pliku sortowanie.bat:

cd kopie

for file in *.zip; do

 y=\$(echo "\$file" | cut -d'-' -f1)

 m=\$(echo "\$file" | cut -d'-' -f2)

 mkdir -p "Rok: \$y/ Miesiac: \$m"

 mv "\$file" "Rok: \$y/ Miesiac: \$m/"

done}

tree (wyświetla listę katalogów, listę plików z wcięciami w głąb)

```
Kasia@DESKTOP-RHFFRLJ UCRT64 ~  
$ cd kopie  
  
Kasia@DESKTOP-RHFFRLJ UCRT64 ~/kopie  
$ cd  
  
Kasia@DESKTOP-RHFFRLJ UCRT64 ~  
$ bash sortowanie.bat  
  
Kasia@DESKTOP-RHFFRLJ UCRT64 ~  
$ tree
```

```
$ bash sortowanie.bat  
Kasia@DESKTOP-RHFFRLJ UCRT64 ~  
$ tree  
.  
├── kopie  
│   ├── Rok: 2010  
│   │   ├── Miesiac: 05  
│   │   │   ├── 2010-05-22.zip  
│   │   │   ├── 2010-05-24.zip  
│   │   │   └── 2010-05-29.zip  
│   │   ├── Miesiac: 06  
│   │   │   ├── 2010-06-03.zip  
│   │   │   └── 2010-06-08.zip  
│   │   ├── Miesiac: 08  
│   │   │   ├── 2010-08-12.zip  
│   │   │   ├── 2010-08-17.zip  
│   │   │   └── 2010-08-20.zip  
│   └── Rok: 2011  
│       ├── Miesiac: 01  
│       │   └── 2011-01-05.zip  
│       ├── Miesiac: 05  
│       │   ├── 2011-05-17.zip  
│       │   └── 2011-05-22.zip  
│       ├── Miesiac: 06  
│       │   └── 2011-06-26.zip  
│       ├── Miesiac: 07  
│       │   ├── 2011-07-24.zip  
│       │   └── 2011-07-29.zip  
│       ├── Miesiac: 08  
│       │   ├── 2011-08-05.zip  
│       │   ├── 2011-08-06.zip  
│       │   ├── 2011-08-11.zip  
│       │   ├── 2011-08-17.zip  
│       │   ├── 2011-08-18.zip  
│       │   └── 2011-08-24.zip  
│       ├── Miesiac: 09  
│       │   ├── 2011-09-05.zip  
│       │   ├── 2011-09-18.zip  
│       │   ├── 2011-09-21.zip  
│       │   └── 2011-09-22.zip  
│       ├── Miesiac: 10  
│       │   ├── 2011-10-03.zip  
│       │   ├── 2011-10-04.zip  
│       │   ├── 2011-10-05.zip  
│       │   └── 2011-10-16.zip  
│       └── Miesiac: 11  
│           └── 2011-11-14.zip  
└── sortowanie.bat  
118 directories, 271 files  
Kasia@DESKTOP-RHFFRLJ UCRT64 ~  
├── 2023-04-08.zip  
│   └── Miesiac: 05  
│       ├── 2023-05-03.zip  
│       └── 2023-05-31.zip  
│   └── Miesiac: 06  
│       └── 2023-06-23.zip  
│   └── Miesiac: 07  
│       ├── 2023-07-02.zip  
│       ├── 2023-07-03.zip  
│       ├── 2023-07-10.zip  
│       └── 2023-07-23.zip  
│   └── Miesiac: 10  
│       ├── 2023-10-16.zip  
│       └── 2023-10-31.zip  
│   └── Miesiac: 11  
│       └── 2023-11-10.zip  
│   └── Miesiac: 12  
│       ├── 2023-12-07.zip  
│       └── 2023-12-21.zip  
└── Rok: 2024  
    ├── Miesiac: 01  
    │   ├── 2024-01-06.zip  
    │   ├── 2024-01-17.zip  
    │   └── 2024-01-29.zip  
    ├── Miesiac: 03  
    │   ├── 2024-03-10.zip  
    │   ├── 2024-03-26.zip  
    │   └── 2024-03-27.zip  
    ├── Miesiac: 04  
    │   ├── 2024-04-08.zip  
    │   └── 2024-04-11.zip  
    ├── Miesiac: 05  
    │   ├── 2024-05-13.zip  
    │   └── 2024-05-27.zip  
    ├── Miesiac: 06  
    │   └── 2024-06-21.zip  
    ├── Miesiac: 07  
    │   └── 2024-07-14.zip  
    ├── Miesiac: 10  
    │   ├── 2024-10-01.zip  
    │   ├── 2024-10-03.zip  
    │   └── 2024-10-09.zip
```

Zadanie 8 - Galeria dla grafika

nano galeria.html (tworzymy plik galeria i otwieramy edytor nano)

Wpisujemy w nim poniższy kod:


```

GNU nano 8.5          galeria.html
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title> Galeria zdjęć</title>
<style>
* { box-sizing: border-box; }
body {font-family: Arial; margin:0; }
.responsive {padding: 0 6px; float: left; width: 25%;}
.gallery img {width:100%; height:auto;}
.gallery {border: 1px solid #ccc;}
.desc {padding: 15px;text-align:center;}
@media only screen and (max-width:700px) {.responsive{width:50%;}}
@media only screen and (max-width:500px) {.responsive{width:100%;}}
.clearfix::after {content: "";display: table; clear:both;}
</style>
</head>
<body>
<h1> Galeria zdjęć</h1>
<p>Kasia Nowakowska i Karolina Spurek</p>
<div class="responsive">
<div class="gallery">
<a target="_blank" href="adrien-olichon-3137064.jpg">

</a>
<div class="desc">adrien-olichon-3137064.jpg</div>
</div>
</div>

```

Jest to tworzenie szkieletu strony html i dodanie css. Zapisujemy i wychodzimy (Ctrl+O, Enter, Ctrl+X).

Następnie w terminalu wpisujemy pętlę for.

```

Kasia@DESKTOP-RHFFRLJ UCRT64 ~/galeria
$ for img in *.jpg; do
> cat <<EOF >> galeria.html
> <div class="responsive">
> <div class="gallery">
> <a target="_blank" href="$img">
> 
> </a>
> <div class="desc">$img</div>
> </div>
> </div>
> EOF
> done

```

Przeszukuje katalog w poszukiwaniu wszystkich plików pasujących do wzorca *.jpg (czyli wszystkich obrazków JPG). Za każdym razem zmienna `img` przyjmuje nazwę jednego z tych plików. Wszystko, co jest między `do` a `done`, będzie wykonane dla każdego pliku osobno.

`cat <<EOF ... EOF` – pozwala podać wieloliniowy tekst bezpośrednio do programu `cat`

>> index.html oznacza, że tekst, który wytworzy cat, zostanie dopisany na końcu pliku galeria.html. Dzięki temu do pliku HTML dopisywany jest za każdym razem cały blok kodu HTML dla pojedynczego obrazka. W tym bloku HTML są użyte zmienne bashowe (np. \$img), które są rozwijane – czyli wstawiana jest faktyczna nazwa pliku obrazu:

- w href="\$img" → link do pełnego rozmiaru zdjęcia,
- w src="\$img" → źródło obrazka w galerii,
- w alt="\$img" i <div class="desc">\$img</div> → nazwa pliku jako podpis pod obrazkiem.

```
echo '<div class="clearfix"></div></body></html>' >> galeria.html
```

{echo '...' >> galeria.html – po prostu dodaje ostatni fragment HTML do pliku;

<div class="clearfix"></div> – element CSS do „czyszczenia” floatów, żeby kontener objął wszystkie elementy .responsive.

</body></html> – zamyka dokument HTML.

>> oznacza dopisanie na końcu pliku, a nie nadpisanie.}

