Project Assignments

Procedure + Projects

Procedure

- Team of 1-2 students
- Several teams can get the same project assignment
- You present your solution in a 5-minute talk
  - So, think of not more than 2-3 slides
- You show:
  - Running application
  - Decomposition: which unit of work? Where did you remove blocking calls?
  - The “core async / parallel code” for implementing the assignment

Procedure

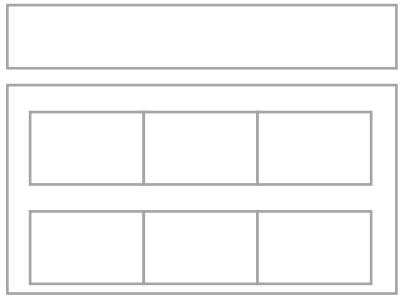
- You send me until April the 5th ☺ EOB: Team member(s) + selected project to birngruber@live.com
- At presentation day, you send me / or share on github / fileshare your project
- If per email:
  - until presentation day, 19:00 o'clock:
  - Subject: JKU-Parallel {Project Name}
  - Body: Team members: Name + Student Ids
  - Attachments / Download folder: Description of your solution + zipped code + slides
- Else (per github / fileshare):
  - Email until presentation day with Link and team members
  - Readme (description) + slides + code is on github / fileshare

Projects

- “GREP”
  - Command line utility grep which accepts wild cards for searching text files and switches –r and -t; e.g. Grep –r –t:100 connect *.txt
  - -r: recursion → so search in folder + subfolders
  - -t: stop after the first n results (order by found file names)

Projects

Image Processing Tool



- MAUI / WPF application, where I can enter a OneDrive folder with pictures, or a Picassa / Pinterest / …. location, or …. – your choice
- Picture thumbnails (128x128) are loaded and shown

- Users can apply filters and transformations to images
- Users can save transformed images
- e.g.: resize all pictures in a folder and save a copy of the transformed pictures…

- UI remains responsive
- I can cancel operations (loading, transformations)
- Think of folders with about > 500 pictures

Projects

- File Uploader (One Person Team)
  - An application (Web, MAUI / WPF, cmd line) that allows users to upload files asynchronously.
  - Use .NET Tasks to handle file I/O operations without blocking the main thread, providing a smooth user experience.
  - Think of upload folders with about > 500 larger files (e.g. your picture collection)

Projects

HDD Tree Size

- MAUI / WPF application
- I want to select a Folder (e.g. C:\)
- Program shows all subfolders + their size in MB
- Program shows all direct files + their size
- I can drill down:
  - select a folder and then I see the sizes again
- UI remains responsive – updates sizes / progress report
- I can cancel and select a new “start folder” any time

Projects

Social Media Feed Aggregator

- MAUI / WPF application for showing live information of different social media feeds / news ... sources
- App fetches data and / or parses web pages
- I can configure data sources (if needed)
- I can see updates in real time
- Think of handling many entries in the app, when app runs for days
  → rolling data?

Projects

- Weather Data Aggregator
  - A service that collects weather data from various APIs concurrently
  - The application should provide real-time weather updates
  - and historical data analysis

Projects

Ping-List-Bot command line tool (max 1 person teams)

- I can provide a list of IP addresses which are pinged
- Tool shows ping result
- Tool continues to ping an IP address
- After first successful ping, tool tries HTTP request on port 80 / 8080 and shows summary: HTTP port 80 / 8080 is open / not open

- Disclaimer: very easy to change this to a DoS HTTP attack bot – please don‘t do this!

Projects

ILoggingService

- I can use ILoggingService { void AddLog(LogEntry entry); } + implementation
  - LogEntry { LogSource (string), EventTime (DateTime), Message (string) }
  - ILoggingService call returns immediately → delegates writing of an entry to a DbLogChannel (consumer)
  - Log method can be called concurrently; LogEntrys are stored in call order for each source
- DbLogChannel writes in a batch: one DB-Connection, one transaction
- DbChannel inserts 1- max 100 records per „transaction insert batch“
- Create a stress test app which adds log entries from 1- N (max 100) concurrent log sources (every 10-x (max 500) milliseconds, a new log entry is added;

Projects

- Real-time Stock Data Processing System
  - Develop a system that processes data in real-time from multiple sources
  - This could simulate a stock ticker processing system, which handles data for a different stock.
  - Alert, e.g., when a rapid price changes
  - App could do concurrent data analyses (like moving averages for multiple stocks at the same time)
  - User friendly interface

Projects

- ETL (Extract, Transform, Load) Pipeline
- Design ETL pipeline(s) that can perform
  - data extraction,
  - transformation,
  - and loading operations in parallel
- This project should demonstrate how to handle big data workflows efficiently

Your Projects … ?

- YOUR project ideas

	Team Red	Team Green	Team Blue
Grep			
Image Processing	Sebastian Hollersbacher	Marcel Ispanki	Ipek Gökaltun
File Uploader	Jonas Brandl		
HDD Tree Size	Dominik Haider		
Social Media Feed Aggregator	Martin Seidl		
Weather Data Aggregator	Simon Buchinger, Niko Sami Diaz Cajas	Kyrylo Meliushko, Viktoriia Kotliarevska	
Ping List Bot	Dominik Mascherbauer	Paul Blume	Oleksandr Bakanov, Yaroslav Kapeliushnyi
ILoggingService			
Realtime Stock Data Processing System			
ETL Pipelines	Zeeshan Nazar (zeeshan.nazar AT protonmail.com) + ???		
MySimpleDB	Fabian Bleck, David Fankhauser		
Near Duplicates Image Tool	Lisa Krimbacher, Martin Dallinger		
Fractal Visualization	Hofstadler Andreas		
Sozial Media Data Analysis (Twitch)	Sebastian Peinbauer, Quirin Ecker		
Unity: Crowd (Action Talking / Fighting)	Dmytro Symovonyk, Vsevolod Kharchenko		
Unity: Inventory + Level Mgmt	Dmytro Romankin		