Operating Systems
INF333

TP03
Design Document & Git

Eda Bahar 26/02/2026
ebahar@gsu.edu.tr
edaabahar@gmail.com

About this TP

- In this TP, you will be learning:
  - Design Document, Software Design Document (SDD)
  - git
  - git for pintos

Design Document

Definition

- A design document is a detailed plan for building a software system. It's like a recipe for your software, laying out all the components and processes needed to create the final product.

- The main goal of a software design document is to turn big-picture ideas into a concrete plan. It helps bridge the gap between what the software should do and how it will be built. By clearly describing the system's structure and features, design documents ensure that all team members are on the same page about what they're building and how they'll do it.

- For more: https://www.atlassian.com/work-management/knowledge-sharing/documentation/software-design-document

Design Document

Definition

- **A good design document usually includes:**
  - **Introduction and overview:** A summary of the system, its purpose, and key objectives, goals and non-goals.
  - **Assumptions and dependencies:** Lists external factors the system relies on, such as third-party services or hardware.
  - **System architecture:** Describes the high-level structure, including principal components and their interactions.
  - **Data design:** Defines how data is structured, stored, and managed within the system.
  - **Interface design:** Details how different system components communicate, including APIs and external connections.
  - **Component design:** Specifies the internal details of individual modules and their responsibilities.
  - **User interface design:** Outlines the look and feel of the system's UI, including layouts and user interactions.
  - **Glossary of terms:** Defines key terms and acronyms used in the document for clarity.

Design Document

Benefits

- Better communication among team members
- Improved project planning and management
- Easier code maintenance and scalability

Design Document

- Example: https://www.bellevuecollege.edu/wp-content/uploads/sites/135/2019/04/SDD_RoadTrip.pdf

HW Design Document

Preliminaries

- Q1: If you have any preliminary comments on your submission, notes for the TAs, or extra credit, please give them here.
    - Example answer: “We added an additional feature to improve thread scheduling efficiency, which is described in comments. Our implementation has extra debugging output that can be ignored.”
- Q2: Please cite any offline or online sources you consulted while preparing your submission, other than the Pintos documentation, course text, and lecture notes.
    - “In addition to the Pintos documentation, the KAIST EE415 Introduction to Operating Systems lecture videos were utilized to gain a better understanding of the project architecture and its requirements. A GitHub repository referenced general ideas on implementing the priority donation but did not copy any code.”

Introduction to git

git

- Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.
  - It allows multiple developers to collaborate efficiently.
  - Helps in managing different versions of a project.
- Why git?
  - It tracks changes and allows reverting if needed.
  - Supports collaboration with branches.
  - Works locally and integrates with remote repositories like GitHub.

Introduction to git

git commands

```
git init # setting up repository

git status # checking status

# adding changes
git add file.txt
git add .
git add -A

# committing changes
git commit -m "initial commit"
git commit -am "add and commit"

# viewing history
git log
git log --online --graph
```

Introduction to git

git commands

```
# creating and switching branches
git branch feature/xyz
git checkout feature/xyz
git switch feature/xyz

# Merging
git merge feature/xyz

# remote
git remote add origin <repo-url>

# pushing and pulling
git push origin main
git pull origin main

# squashing
git rebase -i HEAD~4

# amending
git commit --amend -m "Updated commit message"

# undoing
git reset --hard <commit-hash>
```

Git

Git for pintos

```
$ git clone https://burakarslan.com/git/pintos.git
Cloning into 'pintos'...
$ cd pintos
$ git remote add origin https://github.com/edaabahar/pintos.git
$ git push -u origin master
Enumerating objects: 638, done.
Counting objects: 100% (638/638), done.
Delta compression using up to 48 threads
Compressing objects: 100% (548/548), done.
Writing objects: 100% (638/638), 5.32 MiB | 49.48 MiB/s, done.
Total 638 (delta 82), reused 638 (delta 82), pack-reused 0
remote: Resolving deltas: 100% (82/82), done.
To https://github.com/edaabahar/pintos.git
 * [new branch]      master -> master
branch 'master' set up to track ‘origin/master'.
```

Git

Git for pintos

```
$ git pull
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 332 bytes | 332.00 KiB/s, done.
From /tmp/212/pintos
   79574e9..e291cbe  master     -> origin/master
Auto-merging src/Makefile
CONFLICT (content): Merge conflict in src/Makefile
Automatic merge failed; fix conflicts and then commit the result.
```

```
<<<<<<< HEAD
# This is going to create a conflict
=======
# Added this in directory 2
>>>>>>> e291cbe622aed15b1c9cc568c4fe8a8a6e3d684f
```