|  |
| --- |
| **開発演習**  **COMET Simulator System**  **プロジェクト計画書** |

|  |  |
| --- | --- |
| 文書番号 | 10031-EXE-001 |
| 版数 | 第1.0版 |
| 発行日 | 2024年06月24日 |
| 作成者 | 舩橋 優佑 |
| 承認日 |  |
| 承認者 |  |

改版履歴

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版　数 | 改版日 | 改版者 | 改版内容 | ページNo. |
| 第1.0版 | 2024年06月24日 | 舩橋 優佑 | 初版作成 | - |

**目　次**

[1. はじめに 4](#_Toc170131067)

[2. 目標 4](#_Toc170131068)

[3. 要求事項管理 5](#_Toc170131069)

[4. 開発計画 5](#_Toc170131070)

[4.1 フェーズ定義 5](#_Toc170131071)

[4.2 開発スケジュール 6](#_Toc170131072)

[4.3 進捗確認 6](#_Toc170131073)

[4.4 開発検証（成果物のレビュー） 7](#_Toc170131074)

[5. 文書管理 7](#_Toc170131075)

[6. 記録管理 7](#_Toc170131076)

# はじめに

本計画書は開発演習の実施計画を定めた文書である。

本演習は仮想コンピュータCOMETⅡを実際のコンピュータ(Linux)上で実現するシミュレータの開発を目的とする。

# 目標

* Ｃ言語で大規模なプログラムを作成できるようになる。
  + 構造体と構造体配列、構造体ポインタを使うことができる。
  + 構造体を使ったテーブル操作を行うことができる。
  + C言語標準ライブラリを使いこなすことができる。（ただし、**getopt関数、およびalloc系関数は使用禁止**）
  + システムコールを使うことができる。
  + includeファイルを作成することができる。
* Linuxコマンド、シェルを使いこなせるようになる。
  + 各種コマンド、ツール、シェルを使いこなし、効率的に作業することができる。
  + プログラムソースの管理ができる。（最新のソースを過不足なく管理・提出できること。）
  + 実習環境でのディレクトリ構成について、何をどのように配置すれば効率的か、よく考えてみましょう。
  + makeを使って効率的にコンパイルを行うことができる。
* ソフトウェア開発手法を理解する。
  + システム要求定義
    - システム要求定義書を理解し、諸機能をモジュール（関数）に分割することができる。
    - テストケースをブラックボックステスト観点で抽出し、どのような条件でどのように確認を行うのかを決定できる（結合テスト仕様書の作成）。
  + ソフトウェア詳細設計
    - 適切な関数インタフェース（関数パラメタ、復帰値）を設計し、関数毎に仕様を設計することができる（ソフトウェア詳細設計書の作成）。
    - テストケースを抽出し、どのような条件でどのように確認を行うのかを決定できる（ソフトウェアユニットテスト仕様書の作成）。
  + プログラミング
    - ソフトウェア詳細設計書とコーディング規約に従い、プログラムを作成できる。
    - ソフトウェアユニットテスト仕様書に従ってテストを実施し、実施記録を残すことができる（ユニットテスト報告書の作成）。
  + 結合テスト
    - 結合テスト仕様書に従ってテストを実施することができる（結合テスト報告書の作成）。
* コンピュータの動作原理を理解する。
  + 機械語（命令）の解読方式を理解できること。
  + フラグレジスタの参照や変更を理解できること。
  + プログラムレジスタの操作を理解できること。
  + アドレッシングモードを理解できること。
  + エラー・例外ケースへの対応が考慮されていること。
* 品質マネジメントシステムの要求事項に従って、開発を進めることができる。
  + プロジェクト計画書を作成することができる。
  + プロジェクト計画書に従って作業し、進捗を管理することができる。
  + 要求事項を管理することができる。
  + 仕様上不明確な点は、講師に確認し（提案し）、承認を得ること。
  + 文書を適切に管理することができる。最新版と旧版を混在管理しないこと。
  + 記録を適切に管理することができる。

# 要求事項管理

要求仕様書上で不明確な項目は、講師に確認する。

確認した内容は書面で記録し、講師の承認を得る。

# 開発計画

## フェーズ定義

SLCPにもとづき、ソフトウェア設計、プログラミング、結合テストを開発フェーズとする。

各開発フェーズのインプット、アウトプットの定義は下記の通りである。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 文書管理番号 |  | フェーズ名称 | | | | | |
| ソフトウェア設計 | | プログラミング | | 結合  テスト | |
| IN | OUT | IN | OUT | IN | OUT |
| 10031-EXE-002 | システム要求定義書 | ○ |  |  |  |  |  |
| 10031-EXE-003 | ソフトウェア詳細設計書 |  | ○ | ○ |  |  |  |
| 10031-EXE-004 | ソフトウェアユニットテスト仕様書（※１） |  | ○ | ○ |  |  |  |
| 10031-EXE-005 | プログラムソース |  |  |  | ○ | ○ |  |
| － | ソフトウェアユニットテスト報告書（※２） |  |  |  | ○ |  |  |
| 10031-EXE-006 | 結合テスト仕様書  （※１） |  | ○ |  |  | ○ |  |
| － | 結合テスト報告書  （※２） |  |  |  |  |  | ○ |

※１：テスト一覧、テスト方法を記述した文書（テスト仕様書）

※２：テスト仕様書にテスト結果を記入／添付した報告書（テスト記録）および不具合管理表

## 開発スケジュール

全体工程表などに開発スケジュールを記述し、管理する。

## 進捗確認

日々の夕会で、講師と進捗確認を行う。確認した内容は議事録に残すこと。

実績をガントチャートに記入し、予定と実績の評価を行う。

## 開発検証（成果物のレビュー）

各開発フェーズについて、各アウトプットのレビューを行う。

レビューの指摘点と修正内容は、各アウトプットに対応したレビュー仕様書に記録する。

|  |  |  |  |
| --- | --- | --- | --- |
| フェーズ | 対象 | 実施時期 | 実施者 |
| ソフトウェア 設計 | ソフトウェア詳細設計書  結合テスト仕様書  ソフトウェアユニットテスト仕様書 | 一部完成時  完成時  一部完成時  完成時  一部完成時  完成時 | 自身、講師  自身、講師  自身、講師  自身、講師  自身、講師  自身、講師 |
| プログラミング | プログラムソース  ソフトウェアユニットテスト報告書 | 完成時  完成時 | 自身、講師  自身、講師 |
| 結合テスト | 結合テスト報告書 | 完成時 | 自身、講師 |

フェーズを終了し次のフェーズに進む場合は、以下の点を確認し、講師の許可を得る。

* 全てのアウトプットが完成している
* レビュー時の指摘事項が全て解決されている

# 文書管理

文書の一覧は、文書管理台帳で管理する。

文書管理番号の付与ルールは以下の通りとする。

10031-EXE-999

999：連番

文書は使用する前に、発行する。

発行時は、文書の体裁（誤字、脱字などがないこと）を確認する。

# 記録管理

管理対象の記録及び管理担当者は、記録管理一覧に記述する。

以上