

電話交換機シミュレータ 解説

創価大学 4 年生 堀田 祐輝

2025 年 7 月 9 日

1 制作動機

ソフトウェア開発の講義の課題として、本システムを制作しました。また、ソフトウェア開発の一環として作成するため、無作為で選ばれた 4 人で本システムを制作しました。

2 使用ツール

本システムの作成に使用した開発環境は VisualStudioCode です。理由としては、コードの構成が色で分かりやすく、作成しやすい環境であると判断したこと、また作者自身が VisualStudioCode を最も使用していた開発環境であるため、選択しました。

使用言語は C です。理由は課題として指定された言語であるため、名義上はクライアントの希望として指定された言語です。

開発環境は Windows10 で、実装環境は Linux です。開発環境に Windows10 を選んだ理由は現在作者の手に持っているのが Windows10 のみだったためです。

実装環境に Linux を選んだ理由としては、課題で指定された環境であるため、こちらも名義上はクライアントの希望として指定されたものです。

3 システムの起動方法

Linux のコマンドプロンプトにて PhoneExecutaFolder 実行ファイルまで ChangeDirectory し、全ての.o ファイルをリンクさせます。
※下記のコードでリンクできます。gcc -o a.out header.h group_ana.o input.o next_state.o num_ana.o or_ana.o output.o pass_ana.o st_ana.o task01.o task04.o task10.o task12.o task14.o task15.o task20.o task23.o task30.o task37.o task40.o task50.o

task54.o task56.o task60.o task63.o task70.o task76.o
ter_ana.o tk_select.o

その後、./a.out を行うことで、実行することが出来ます。

4 システム内容説明

4.1 概要

本システムは電話を行うための電話線のシステムをアルゴリズムにしたものです。

具体的には、黒電話の電話線を交換するシステムをプログラミングで再現したものです。

4.2 システムの流れ

起動すると、初期設定変更を行うことが出来ます。

発信許可設定は設定した端末は他の端末と接続できなくなるものです。

状態変更はデバッグ用に存在し、対応する数値を入力することで、空き状態、ダイヤル状態、呼び出し状態、通話状態、ビジー状態、パス入力状態、グループ通話状態、パス設定状態のいずれかにすることが出来ます。

その後、イベント (0:offhook 1:onhook 2:dial 3:group)>と出力されます。ここで 0~3 の数値を入力することで、対応した処理がされます。

また 1 は本システムの基本となった状態遷移図で、2 は追加した機能を含めた状態遷移図です。それぞれ状態名が存在し、

1. S0：空き状態
2. S1：ダイヤル待ち状態
3. S2：応答待ち状態
4. S3：通話状態
5. S4：ビジー状態
6. S5：パスワード入力待ち状態

- 7. S6：グループ通話状態
- 8. S7：パスワード設定待ち状態

上記の状態が存在します。

4.2.1 offhook (オフフック) の処理

0 はオフフック、受話器を持ち上げる動作をアルゴリズムに落とし込んだ箇所で、入力することで空き状態の端末をダイヤル状態に出来ます。

4.2.2 onhook (オンフック) の処理

1 はオンフック、受話器を置く動作をアルゴリズムに落とし込んだ箇所で、入力することで、全ての状態から空き状態に出来ます。

4.2.3 dial (ダイヤル) の処理

2 はダイヤル、通信したい端末番号を入力することで、端末同士で通信することが出来ます。入力することで、呼び出した端末と呼び出された端末を呼び出し状態にすることが出来ます。しかし、呼び出される端末が通話状態かビジー状態である場合は通信することが出来ずに、呼び出した端末がビジー状態になります。

4.2.4 group (グループ通話) の処理

3 はグループ通話であり、パスワードを設定してグループを作成します。グループを作成した端末と通信しようとするパスワードの入力を要求され、設定したパスワードを入力することでグループに参加して通信することが出来ます。

4.2.5 デバッグ機能

また、i と入力することで、個別のデータ情報を閲覧することが出来、発信拒否している端末、端末ごとの状態の確認、発着識別子(グループ通話か、1対1通話かどうか)の確認、設定されたパスワードの確認を行うことが出来ます。

5 担当箇所

著者が担当した箇所は task23.c, task30.c, task76.c, st_ana.c(デバッグと修正のみ), ter_ana.c, output.c です。

task ○○は画像 2 の状態の番号の遷移を行うコードです。

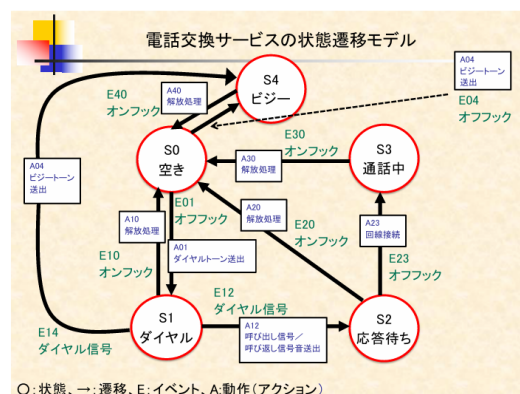


図1 基本となった状態遷移図

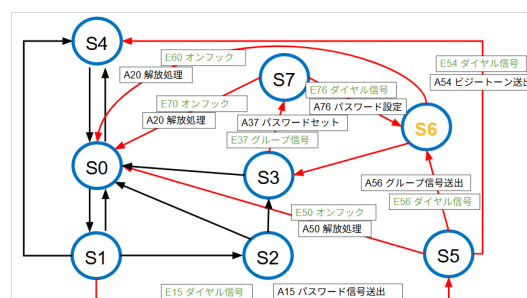


図2 拡張後の状態遷移図

st_ana は引数の端末の状態を確認して値を返すコードです。

ter_ana は引数の端末が通話中・グループ通話中かどうか確認して結果を返すコードです。

output は接続や音の送出をユーザーに伝えるためのコードです。

6 アピールポイント (工夫した点)

本システムの工夫した点は課題で提示された機能の実装の他に、状態を変更できるデバッグ機能、複数人で通話することが出来るグループ通話機能、パスワードを設定することで通話に入る人を制限することができるセキュリティ機能を追加したことです。

追加した理由としては、デバッグ機能は拡張する際、テストとして実行する場合に容易にエラーチェックができるため、追加しました。

グループ通話機能は、より迅速に情報共有することや簡単に複数の人と会話することが非常に有益であると考えたため、追加しました。

パスワード機能はグループ通話中、だれでも簡単にグループに入れると安全性と機密性が保たれず、道の真ん中で大きな声で話している状態と同義であ

り、電話である必要性が低くなるため、追加しました。

また、グループで作成するということで、分かりやすい変数名やコメントを意識して記述しました。

他にも、全く知らない人とも話せるように、趣味の話や最近あった出来事など自分でも話しやすいことから切り出すように意識します。

7 苦戦した点

苦戦した点は、開発チーム全員が追加する機能についての構成や流れを共有することでした。

そもそも追加する前の機能の構成を理解できていない人もいたなど、システムの構成についての理解度に違いがありました。

したがって、状態遷移図を作成して提示したり、話した内容をまとめたものを整理して共有したり、話の区切りで理解できているか確認したりして、チーム全員が把握できるように心がけました。

しかし、最終的に理解が出来ていない事態が発生して、一人一人の仕事量に差が出てしまいました。

主な原因としては、理解できているか確認するときに圧が出てしまったことで話を切り出せなかったり、特定の人物との会話が多くなり、情報共有や構成の理解に支障が発生してしまったりしたことです。

この経験から全員の人が理解と意見が言えるような環境づくりが大切であるということを実感しました。今後は口調や会話の振り方に意識することと、困ったらいつでも話せる環境を整えてからチーム開発に望もうと考えました。

8 本作品にて学習できた点

本プロジェクトを通して、上記のようにチーム開発環境が大切であることの他に、自分はあまり他人を信用していないということも理解しました。

理由としては、仕事を分担するとき、難しい内容を他人に頼むことをためらい、簡単な内容しか頼んでいないことに気が付いたためです。また、特定の人物との会話が多くなることも他の人を信用していないという裏付けであると気づきました。

一方で、全く知らない人を信用するということは個人的に難しいため、今後は失敗してもリスクを軽減できるような保険を作り、「失敗してもあまり損失がない」と考えさせることから始めたいと考えています。