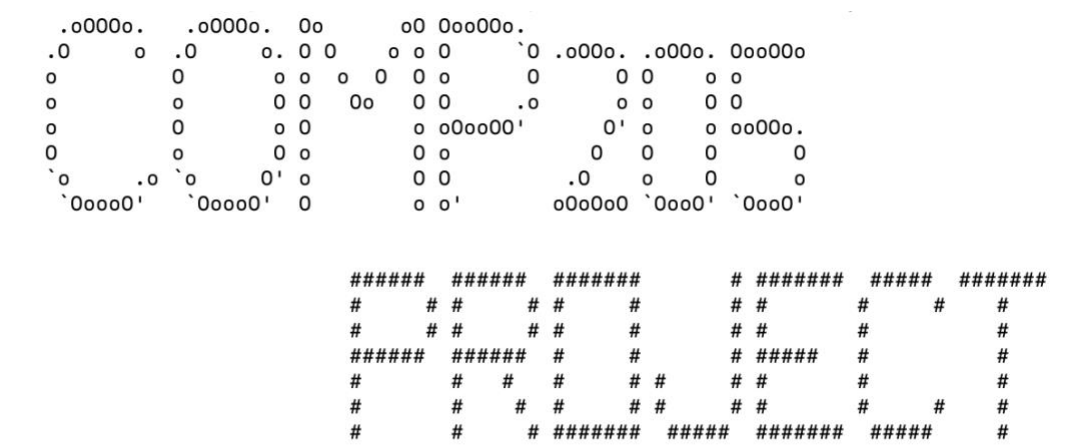
# COMP205 Systems Programming
# Project Guide and Report

Ozan Sönmez
041701015

# Table of contents

# Project Overview

## COMP205 System Programming Project - Screenshot Organizer and Contact Manager




This website has been generated for a walk-through and a report for System Programming Project.
This report inspect one BASH Script and one C programmed file. This Project includes one Screenshot organizer for mac os x products and a contact manager.

This program is designed for Mac OS X CATALINA or upper version(s) users and uses GNU bash, version 5.0.11(1)-release (x86_&4-apple-darwin18.6.0) and not yet tested on Windows platform.

This guide and mentioned code has been prepared under the license CC BY0 1.0 Universal and its free for all.

Requirements: All code files and the executable must be in the same folder

## Bash - Screenshot Organizer For MacOSX

```
cd $HOME/Desktop
[ ! -d "$HOME/Desktop/ScreenShots" ] && mkdir -p "$HOME/Desktop/ScreenShots"
[ ! -d "$HOME/Desktop/ScreenShots/January" ] && mkdir -p "$HOME/Desktop/ScreenShots/January"
[ ! -d "$HOME/Desktop/ScreenShots/February" ] && mkdir -p "$HOME/Desktop/ScreenShots/February"
[ ! -d "$HOME/Desktop/ScreenShots/March" ] && mkdir -p "$HOME/Desktop/ScreenShots/March"
[ ! -d "$HOME/Desktop/ScreenShots/April" ] && mkdir -p "$HOME/Desktop/ScreenShots/April"
[ ! -d "$HOME/Desktop/ScreenShots/May" ] && mkdir -p "$HOME/Desktop/ScreenShots/May"
[ ! -d "$HOME/Desktop/ScreenShots/June" ] && mkdir -p "$HOME/Desktop/ScreenShots/June"
[ ! -d "$HOME/Desktop/ScreenShots/July" ] && mkdir -p "$HOME/Desktop/ScreenShots/July"
[ ! -d "$HOME/Desktop/ScreenShots/August" ] && mkdir -p "$HOME/Desktop/ScreenShots/August"
[ ! -d "$HOME/Desktop/ScreenShots/September" ] && mkdir -p "$HOME/Desktop/ScreenShots/September"
[ ! -d "$HOME/Desktop/ScreenShots/October" ] && mkdir -p "$HOME/Desktop/ScreenShots/October"
[ ! -d "$HOME/Desktop/ScreenShots/November" ] && mkdir -p "$HOME/Desktop/ScreenShots/November"
[ ! -d "$HOME/Desktop/ScreenShots/December" ] && mkdir -p "$HOME/Desktop/ScreenShots/December"
find . -maxdepth 1 -name "*-01-*" >> "$HOME/Desktop/January.txt"
find . -maxdepth 1 -name "*-02-*" >> "$HOME/Desktop/February.txt"
find . -maxdepth 1 -name "*-03-*" >> "$HOME/Desktop/March.txt"
find . -maxdepth 1 -name "*-04-*" >> "$HOME/Desktop/April.txt"
find . -maxdepth 1 -name "*-05-*" >> "$HOME/Desktop/May.txt"
find . -maxdepth 1 -name "*-06-*" >> "$HOME/Desktop/June.txt"
find . -maxdepth 1 -name "*-07-*" >> "$HOME/Desktop/July.txt"
find . -maxdepth 1 -name "*-08-*" >> "$HOME/Desktop/August.txt"
find . -maxdepth 1 -name "*-09-*" >> "$HOME/Desktop/September.txt"
find . -maxdepth 1 -name "*-10-*" >> "$HOME/Desktop/October.txt"
find . -maxdepth 1 -name "*-11-*" >> "$HOME/Desktop/November.txt"
find . -maxdepth 1 -name "*-12-*" >> "$HOME/Desktop/December.txt"

#args -I{} < November.txt  mv -v {} /November  // READ-ONLY ERROR AFTER MAC OS CATALINA
while read i; do    temp="${i%\"}";    temp="${temp#\"}"    cp "$temp" "$HOME/Desktop/ScreenShots/January"; done < January.txt
while read i; do    temp="${i%\"}";    temp="${temp#\"}"    cp "$temp" "$HOME/Desktop/ScreenShots/February"; done < February.txt
while read i; do    temp="${i%\"}";    temp="${temp#\"}"    cp "$temp" "$HOME/Desktop/ScreenShots/March"; done < March.txt
while read i; do    temp="${i%\"}";    temp="${temp#\"}"    cp "$temp" "$HOME/Desktop/ScreenShots/April"; done < April.txt
while read i; do    temp="${i%\"}";    temp="${temp#\"}"    cp "$temp" "$HOME/Desktop/ScreenShots/May"; done < May.txt
while read i; do    temp="${i%\"}";    temp="${temp#\"}"    cp "$temp" "$HOME/Desktop/ScreenShots/June"; done < June.txt
while read i; do    temp="${i%\"}";    temp="${temp#\"}"    cp "$temp" "$HOME/Desktop/ScreenShots/July"; done < July.txt
while read i; do    temp="${i%\"}";    temp="${temp#\"}"    cp "$temp" "$HOME/Desktop/ScreenShots/August"; done < August.txt
while read i; do    temp="${i%\"}";    temp="${temp#\"}"    cp "$temp" "$HOME/Desktop/ScreenShots/September"; done < September.txt
while read i; do    temp="${i%\"}";    temp="${temp#\"}"    cp "$temp" "$HOME/Desktop/ScreenShots/October"; done < October.txt
while read i; do    temp="${i%\"}";    temp="${temp#\"}"    cp "$temp" "$HOME/Desktop/ScreenShots/November"; done < November.txt
while read i; do    temp="${i%\"}";    temp="${temp#\"}"    cp "$temp" "$HOME/Desktop/ScreenShots/December"; done < December.txt
sudo rm Screen*.png
```

**Screenshot Organizer for MacOSX**

MacOSX ,by default, saves the screenshots (CMD+SHIFT+3||4) to the Desktop folder on the devices. This causes a lot of headache when a user takes too many screenshots frequently or never cleans the old one.

This Bash script aims to help the user by handling necessary file operations to clean up the Desktop folder and categorize the screenshots by month.

This code snippet uses a very straightforward approach for handling file operations, the operation scheme is as it follows:

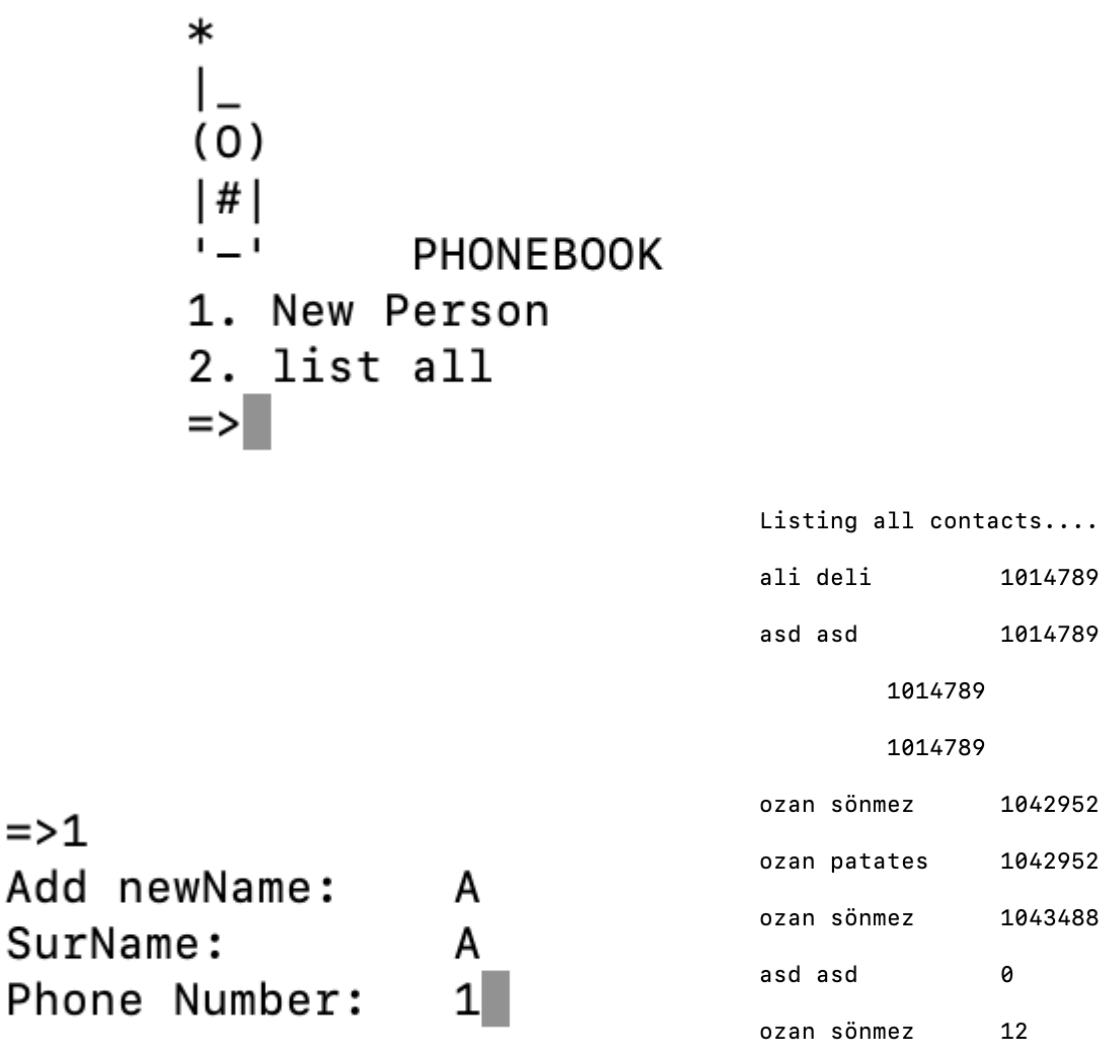
1. System Checks if the ScreenShots folder has been already estabilished. (For users who are using the program more than once)
2. System checks if the folders belonging to months in ScreenShots folder has been already estabilished and creates the necessary folders if some folder is non-existent.
3. MacOSX's default Screen Shot name is as it follows: "Screen Shot YYYY-MM-DD TI:ME:00" Bash function uses 'find' operation to detect months and appends the screen shots' names to the respective month.txt files. This files are temprorary files and later removed by the bash script.
4. System uses While-do loops for copying the Screenshots in the desktop folder to their new and organized place using the .txt files mentioned above.
5. System deletes all the screenshots in the ./Desktop folder
6. System deletes all the .txt files used in this process
7. The code successfully completes its job and return to the main terminal page.

### ATTENTION!

This program was not able to use certain operations to speed up the process because of the Apple's new updates with the Mac OS X CATALINA.

> **If you upgrade to macOS Catalina from an earlier version of macOS, the read-only volume is created during the upgrade process. Files or data that you previously stored in the startup volume are now stored in this new volume, and some of these files may appear in a new folder called Relocated Items. You can check this folder for any files that you can’t locate.**
>
> **With macOS Catalina, you can no longer store files or data in the read-only system volume, nor can you write to the "root" directory ( / ) from the command line, such as with Terminal.**
> **Apple, Published Date: October 12, 2019**

## C - Contact Manager




Contact Manager

This program aims to get users contacts and saving them into a flat .txt file.

The program scheme is as it follows:

1. Users opens up the Contacts from the main terminal
2. Users decides whether if he wants to add a new contact or display all the contacts from the terminal
3. If the users chooses the add a new contact, system calls the addnew.c file and get user's inputs.
4. The addnew() function appends the new contact to the list.txt file.
5. The users gets promted to the main screen
6. If the user selects to list all the contacts, system calls the list.c file
7. listall() function lists all the contacts using the file operands and returns 0;
8. User gets prompted to the main screen

## Sample Execution

### SAMPLE TERMINAL EXECUTION

```
Last login: Wed Dec 25 03:48:24 on ttys001
Ozans-MacBook-Pro:~ mefuniversity$ /Users/mefuniversity/Desktop/Project/someprog ; exit;
                .oOOOo.   .oOOOo.  Oo      oO OooOOo.
               .O     o  .O     o. O O    o o O     `O .oOOo. .oOOo. OooOOo
               o         O       o o  o  O  O o      O      O O    o o
               o         o       O O   Oo   O O    .o      o o    O O
               o         O       o O       o oOooOO'      O' o    o ooOOo.
               O         o       O o       O o           O   O    O      O
               `o     .o `o     O' o       O O         .O    o    O      o
                `OoooO'   `OoooO'  O       o o'       oOoOoO `OooO' `OooO'

                                    ######  ######  #######       # #######  #####  #######
                                    #     # #     # #     #       # #       #     #    #
                                    #     # #     # #     #       # #       #          #
                                    ######  ######  #     #       # #####   #          #
                                    #       #   #   #     # #     # #       #          #
                                    #       #    #  #     # #     # #       #     #    #
                                    #       #     # #######  #####  #######  #####     #


AVAILABLE COMMANDS: clean, contact
clean
Password:
                .oOOOo.   .oOOOo.  Oo      oO OooOOo.
               .O     o  .O     o. O O    o o O     `O .oOOo. .oOOo. OooOOo
               o         O       o o  o  O  O o      O      O O    o o
               o         o       O O   Oo   O O    .o      o o    O O
               o         O       o O       o oOooOO'      O' o    o ooOOo.
               O         o       O o       O o           O   O    O      O
               `o     .o `o     O' o       O O         .O    o    O      o
                `OoooO'   `OoooO'  O       o o'       oOoOoO `OooO' `OooO'

                                    ######  ######  #######       # #######  #####  #######
                                    #     # #     # #     #       # #       #     #    #
                                    #     # #     # #     #       # #       #          #
                                    ######  ######  #     #       # #####   #          #
                                    #       #   #   #     # #     # #       #          #
                                    #       #    #  #     # #     # #       #     #    #
                                    #       #     # #######  #####  #######  #####     #


AVAILABLE COMMANDS: clean, contact
```

### STEPS

1. Users opens up the terminal via the executable file
2. The code prompts up a screen and shows up the available commands
3. User types 'clean'
4. System asks for user password
5. Users enters the password and presses enter
6. System completes the processes in the background and a new terminal window opens up to as for new commands

With this code, The author was not able to use the command 'contact' within the terminal. The related code to contact manager works as expected outside of the project itself but some problems occured during the implementation of the code to the main.c file.