sx004098

Generated by Doxygen 1.8.17

Chapter 1

_FrontPage

Module Code: CS1PC20 Assignment report Title: Autumn Coursework 2 Student Number: 30004098 Date↩ : 11/01/2022 Actual hours spent on the assignment: 20 hours Assignment evalualtion: We clearly weren't taught enough, but it allowed us to test our problem solving

Chapter 2

Coursework 2 Project Planning exercise

Planning is the keystone of successful programming, but it need not all be dry paperwork. The idea of experimenting with code in a series of feasibility studies can really help identify which approach to take.

In this exercise, you have some thinking to do, and you should get together with others in your peer group (other students) to discuss ideas.

First, though, grab a copy of this project by going to it on CSGitLab and using the "Fork" button near the top right corner.

Please set your version of the project to "private" after forking.

2.1 Goals

Identify the key goals you need to complete for the coursework

1. Complete on time
2. Complete all "must" and "should" features
3. Manage time and if possible, make an attempt on "could" features
4. 5.

2.2 Requirements to fulfill goals

What do you need to be able to meet those goals? This can include clearer specifications, tests (what needs testing, how to test it?), knowledge etc.

1. 1. 1. 1. 1.

2.3 Dependencies (mapping goals/requirements etc)

Which requirements relate to which goals - think about drawing out a Product Breakdown for the coursework

1. 1. 1. 1. 1.

Hint: This might look something like this:

```
graph LR

    Project --> 1_must_haves
    subgraph "Must have"
    1_must_haves --> 1_create_folder_structure
    1_must_haves --> 2_abort_if_exists
    1_must_haves --> 3_initialise_git
    1_must_haves --> 4_feature_management
    end

    4_feature_management --> manage_features
    subgraph "manage features"
        manage_features(Manage Features)
        manage_features --> 1_shorthand_codes
        manage_features --> 2_shorthand_lookup
        manage_features --> 3_git_branch_per_feature
    end

    Project --> 2_should_haves
    subgraph "Should have"
    2_should_haves --> 5_renaming_features
    2_should_haves --> 6_moving_features
    2_should_haves --> 7_tree_diagram
    2_should_haves --> 8_time_workload_data
    2_should_haves --> 9_time_workload_on_diagram
    end

    Project --> 3_could_haves
    Project --> 4_wont_haves
    Project --> readme.md
```

2.4 Plan:

2.4.1 What experimental code can I write now? (feasibility studies)

1. 1. 1. 1. 1.

Hint: try modifying the make file created in week4, consider the code in the videos for ideas about automating your code build, and the video on restucturing your project folder, for instance.

2.4.2 What were the results of trying things out?

1. 1. 1. 1. 1.

Chapter 3

README

to start open terminal in this folder and type: source pm.sh

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

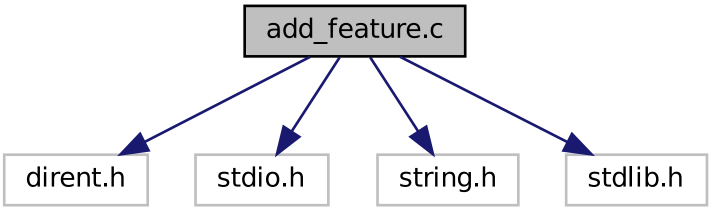
File Documentation

5.1 _FrontPage.md File Reference

5.2 add_feature.c File Reference

```
#include <dirent.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
```

Include dependency graph for add_feature.c:



Macros

- #define _DIR 1
- #define MAX_NUM 100
- #define NAME_LENGTH 100

Functions

- int list_file_type (char type, char result[MAX_NUM][NAME_LENGTH])
- int main (void)

Variables

- int dup =0

5.2.1 Macro Definition Documentation

5.2.1.1 _DIR

```
#define _DIR 1
```

5.2.1.2 MAX_NUM

```
#define MAX_NUM 100
```

5.2.1.3 NAME_LENGTH

```
#define NAME_LENGTH 100
```

5.2.2 Function Documentation

5.2.2.1 list_file_type()

```
int list_file_type (
            char type,
            char result[MAX_NUM][NAME_LENGTH] )
```

5.2.2.2 main()

```
int main (
            void )
```

5.2.3 Variable Documentation

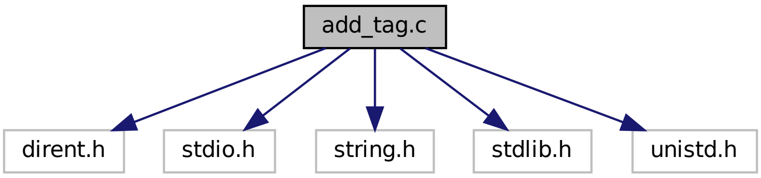
5.2.3.1 dup

```
int dup =0
```

5.3 add_tag.c File Reference

```
#include <dirent.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
```

Include dependency graph for add_tag.c:



Macros

- #define _FILE 1
- #define MAX_NUM 100
- #define NAME_LENGTH 100

Functions

- int list_file_type (char type, char result[MAX_NUM][NAME_LENGTH])
- int main (void)

Variables

- int duped =0

5.3.1 Macro Definition Documentation

5.3.1.1 _FILE

```
#define _FILE 1
```

5.3.1.2 MAX_NUM

```
#define MAX_NUM 100
```

5.3.1.3 NAME_LENGTH

```
#define NAME_LENGTH 100
```

5.3.2 Function Documentation

5.3.2.1 list_file_type()

```
int list_file_type (
             char type,
             char result[MAX_NUM][NAME_LENGTH] )
```

5.3.2.2 main()

```
int main (
             void  )
```

5.3.3 Variable Documentation

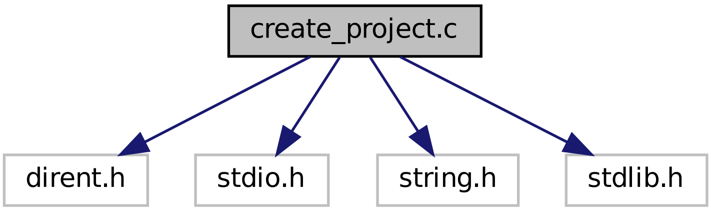
5.3.3.1 duped

```
int duped =0
```

5.4 create_project.c File Reference

```
#include <dirent.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
```

Include dependency graph for create_project.c:



Macros

- #define _DIR 1
- #define MAX_NUM 100
- #define NAME_LENGTH 100

Functions

- int list_file_type (char type, char result[MAX_NUM][NAME_LENGTH])
- int main (void)

Variables

- int dup =0

5.4.1 Macro Definition Documentation

5.4.1.1 _DIR

```
#define _DIR 1
```

5.4.1.2 MAX_NUM

```
#define MAX_NUM 100
```

5.4.1.3 NAME_LENGTH

```
#define NAME_LENGTH 100
```

5.4.2 Function Documentation

5.4.2.1 list_file_type()

```
int list_file_type (
            char type,
            char result[MAX_NUM][NAME_LENGTH] )
```

5.4.2.2 main()

```
int main (
            void  )
```

5.4.3 Variable Documentation

5.4.3.1 dup

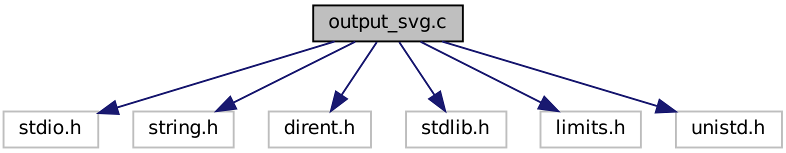
```
int dup =0
```

5.5 exercise7.md File Reference

5.6 output_svg.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <dirent.h>
#include <stdlib.h>
#include <limits.h>
#include <unistd.h>
```

Include dependency graph for output_svg.c:



Functions

- void listfiles (const char ∗dirname, int level)
- int draw_list_files (char ∗name)
- int main (int argc, char ∗argv[ ])

5.6.1 Function Documentation

5.6.1.1 draw_list_files()

```
int draw_list_files (
            char * name )
```

5.6.1.2 listfiles()

```
void listfiles (
            const char * dirname,
            int level )
```

5.6.1.3 main()

```
int main (
            int argc,
            char * argv[] )
```

5.7 README.md File Reference

Index