CS232 Operating Systems
Assignment 02: Introduction to System Calls

Fatima Nadeem (fn03768)   Muhammad Shahrom Ali (ma03559)

Fall 2019

1 Question No. 1

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

int main(int argc, char *argv[])
{
        if (argc == 1) //Just ./mycat
        {
                do
                {
                        char c = fgetc(stdin);

                        if (feof(stdin))
                        {
                                break;
                        }

                        printf("%c", c);
                } while (1);
        }

        else if (argc >= 2)
        {
                if (strcmp(argv[1], "-n") != 0) //If there is -n
                {
                        for (int i = 1; i < argc; i++)
                        {
                                FILE* fp;

                                fp = fopen(argv[i], "r");
                                if (fp == NULL)
                                {
                                        fprintf(stderr, "mycat:␣%s:␣No␣such␣file␣
                                        exit(1);
                                }

                                do
                                {
                                        char c = fgetc(fp);
                        if (feof(fp))
                        {
                                break;
                        }

                        printf("%c", c);
                } while (1);

                fclose(fp);
        }
}
else //If there is no -n
{
        int ind = 1;

        for (int i = 2; i < argc; i++)
        {
                FILE* fp;

                fp = fopen(argv[i], "r");
                if (fp == NULL)
                {
                        fprintf(stderr, "mycat:␣%s:␣No␣such␣file␣
                        exit(1);
                }

                int filestrt = 1;

                do
                {
                        char c = fgetc(fp);

                        if (feof(fp))
                        {
                                break;
                        }

                        if (filestrt == 1)
                        {
                                printf("%d\t", ind);
                                filestrt = 0;
                        }

                        printf("%c", c);

                        if (c == '\n')
                        {
                                ind++;
                                printf("%d\t", ind);
                        }
                } while (1);

                fclose(fp);
        }
}
```

```
            }
    }
```

2 Question No. 2

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <sys/wait.h>

//Prototypes
int string_length(char* );
void print_string_array(char* const array[]);

struct process
{
        int pid;
        char* name;
        int status;
};

//Main
int main(int argc, char const *argv[])
{
        printf("\033cGreetings␣Master\n");

        //Setting Path variable data structure
        char *VAR[16];
        int j = 0;
        VAR[j] = strtok(getenv("PATH"), ":");
        while (VAR[j] != NULL)
        {
                j++;
                VAR[j] = strtok(NULL, ":");
        }
        VAR[j] = "./";
        j++;
        VAR[j] = NULL;


        int Exit = 0;
        char* prompt = "Master@HUSh:␣Your␣wish␣is␣my␣command$␣";
        int max_command_length = 255;

        while (!Exit)
        {
                printf("%s", prompt);

                char* command = malloc(max_command_length);

                command = fgets(command, max_command_length, stdin);

                if (command == NULL)
                {
                        fprintf(stderr, "ERROR!␣With␣all␣due␣respect␣Master,␣what
                }
```

```
else
{

        if (strncmp("exit", command, 4) == 0 && string_length(com
        {
                //Kill Running processes

                printf("So␣be␣it,␣Master.␣I␣will␣wait␣for␣your␣re
                Exit = 1;
        }
        else if (strncmp("clear", command, 5) == 0 && string_leng
        {
                printf("\033c");
        }

        else
        {
                int new_proc = fork();

                if (new_proc < 0)
                {
                        fprintf(stderr, "Invalid␣Fork␣master!\n")
                        free(command);
                        exit(1);
                }

                else if (new_proc == 0)
                {

                        int no_of_arg = 100;
                        char* child_argv[no_of_arg];
                        int j = 0;

                        child_argv[j] = strtok(command, "␣");

                        while (child_argv[j] != NULL)
                        {
                                j++;
                                child_argv[j] = strtok(NULL, "␣")
                        }
                        child_argv[j-1] = strtok(child_argv[j-1],
                        child_argv[j] = NULL; //END OF ARRAY

                        char buffer[255];
                        int v = 0;

                        //Check all directories in path to run in
                        while (execvp(buffer, child_argv) == -1)
                        {
                                strcpy(buffer, VAR[v]);
                                strcat(buffer, "/");
                                strcat(buffer, child_argv[0]);
                                // printf("%s\n", buffer);
                                v++;
                        }
```

```
                                        fprintf(stderr, "FAILED␣to␣load␣%s\n", ch
                                        printf("\nI␣dont␣understand␣your␣Wish,␣ma
                                        free(command);
                                        exit(-1);
                                }
                                else
                                {
                                        //PARENT
                                        int termed_proc = wait(NULL);
                                }
                        }

                }
                free(command);
        }
        return EXIT_SUCCESS;
}


int string_length(char* string)
{
        int i = 0;
        while (string[i])
        {
                i++;
        }
        return i-1;
}

void print_string_array(char* const array[])
{
        int j = 0;
        while (array[j] != NULL)
        {
                printf("%s\n", array[j]);
                j++;
        }
}
```

3 Comments

loved it! A bit tough given the current course load but loved it all the same!

4 Appendix A: Makefile

```
compileall: mycat hush

mycat: Lec1_gp05_A2Q1_mycat.o
        gcc Lec1_gp05_A2Q1_mycat.o -o mycat

hush: Lec1_gp05_A2Q2_Hush.o
        gcc Lec1_gp05_A2Q2_Hush.o -o hush

Lec1_gp05_A2Q2_Hush.o: Lec1_gp05_A2Q2_Hush.c
        gcc -c Lec1_gp05_A2Q2_Hush.c

Lec1_gp05_A2Q1_mycat.o: Lec1_gp05_A2Q1_mycat.c
        gcc -c Lec1_gp05_A2Q1_mycat.c

wrap:
        tar -zcvf Lec1_gp05_A2.tar.gz ./Makefile ./*.c ./Lec1_gp05_A2.pdf ./*.te

clean:
        rm *.o mycat hush
```