UCSC Extension – Linux System Programming
Homework #2
Jae Yang Park (jaeyangp@gmail.com)

2.1

```
// Homework #2
// Q 2.1

#define _POSIX_SOURCE 1

#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <errno.h>
#include <string.h>


int main(int argc, char **argv)
{
        int newfd;
        int nchar = 0;
        int nword = 0;
        int nline = 0;
        int c;

        if (argc == 1) {
                fprintf(stderr, "No input text file!\n");
                exit(1);
        };

        char *rfn = argv[1];

        // stdin redirection
        close(0);
        newfd = open(rfn, O_RDONLY);

        if (newfd < 0) {
                fprintf(stderr, "Error: open `%s' failed: %s\n", rfn, strerror(errno));
                exit(1);
        }

        while ((c = getchar()) != EOF) {
                //putchar(c);
                nchar++;

                if (c == ' ')
                        nword++;

                if (c == '\n')
                        nline++;
        }

        printf("\n%d characters, %d words, and %d lines in the %s!\n\n", nchar, nword, nline, rfn);

        close(newfd);
        exit(0);
}
```

2.2 a, b

```
// Homework #2
// Q 2.2

#define _POSIX_SOURCE 1
```

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <errno.h>
#include <string.h>
#include <unistd.h>


int main(int argc, char **argv)
{
        int newfd;
        char *cp;
        size_t nc;

        if (argc == 1) {
                fprintf(stderr, "No input text file!\n");
                exit(1);
        };

        char *rfn;
        int i;

        // stdin redirection
        close(0);

        for (i = 0; i < (argc-1); i++) {
                rfn = argv[i+1];

                newfd = open(rfn, O_RDONLY);

                if (newfd < 0) {
                        fprintf(stderr, "Error: open `%s' failed: %s\n", rfn, strerror(errno));
                        exit(1);
                }

                cp = (char *)malloc(sizeof(char));

                while ((nc = read(newfd, cp, 1))) {
                        putchar(*cp);
                }

                close(newfd);
                free(cp);
        }

        exit(0);
}

// Homework #2
// Q 2.2B

#define _POSIX_SOURCE 1

#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <errno.h>
#include <string.h>


int main(int argc, char **argv)
{
```

```
        int newfd;
        int c;

        if (argc == 1) {
                fprintf(stderr, "No input text file!\n");
                exit(1);
        };

        char *rfn;
        int i;
        int line_flag = 0;

        // stdin redirection
        close(0);

        for (i = 1; i < argc; i++) {
                if (!strcmp(argv[1], "-l")) {
                        if (i > (argc-2))
                                break;

                        rfn = argv[i+1];
                        line_flag = 1;
                } else {
                        rfn = argv[i];
                }

                newfd = open(rfn, O_RDONLY);

                if (newfd < 0) {
                        fprintf(stderr, "Error: open `%s' failed: %s\n", rfn, strerror(errno));
                        exit(1);
                }

                if (line_flag) {
                        printf("# ");
                        while ((c = getchar()) != EOF) {
                                if (c == '\n') {
                                        putchar('\n');
                                        putchar('#');
                                        putchar(' ');
                                } else
                                        putchar(c);
                        }
                } else {
                        while ((c = getchar()) != EOF) {
                                putchar(c);
                        }
                }

                close(newfd);
        }

        putchar('\n');

        exit(0);
}

2.3
// Q2.3

#include <stdio.h>
#include <unistd.h>

int main()
{
        printf("Maximum number of open files = %ld\n", sysconf(_SC_OPEN_MAX));
```

```
        return 0;
}

2.4
// Homework #2
// Q 2.4

#define _POSIX_SOURCE 1

#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <errno.h>
#include <string.h>
#include <unistd.h>

int main(int argc, char **argv)
{
        int newfd, newfd1;
        int c;

        if (argc == 1) {
                fprintf(stderr, "No input text file!\n");
                exit(1);
        };

        char *rfn = argv[1];
        char *wfn = "my_new_file.txt";

        newfd = open(rfn, O_RDONLY);

        if (newfd < 0) {
                fprintf(stderr, "Error: open %s failed\n", rfn);
                exit(1);
        }

        if (dup2(newfd, 0) != 0) {
                fprintf(stderr, "Error: could not dup2 %s\n", rfn);
                exit(1);
        }

        close(newfd);

        newfd1 = open(wfn, O_CREAT | O_WRONLY | O_TRUNC, S_IRUSR | S_IWUSR | S_IRGRP | S_IROTH);

        if (newfd1 < 0) {
                fprintf(stderr, "Error: open %s failed\n", wfn);
                exit(1);
        }

        if (dup2(newfd1, 1) != 1) {
                fprintf(stderr, "Error: could not dup2 %s\n", wfn);
                exit(1);
        }

        close(newfd1);

        int lc = 0;
        while ((c = getchar()) != EOF) {
                putchar(c);

                if (c == '\n')
                        lc++;
        }
```

```
        fprintf(stderr, “%d lines copied\n”, lc);

        close(0);
        close(1);
        exit(0);
}
```

2.5 a

```
// Q 2.5A

#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <dirent.h>
#include <string.h>

int main(int argc, char **argv)
{
        DIR *dirp;
        struct dirent *dp;
        char *dir_name;

        dir_name = ".";

        dirp = opendir(dir_name);

        if (dirp == NULL) {
                fprintf(stderr, "Error: can't open '%s'\n", dir_name);
                exit(1);
        }

        while ((dp = readdir(dirp)) != NULL) {
                if (strcmp(dp->d_name, ".") == 0 || strcmp(dp->d_name, "..") == 0)
                        continue;
                else
                        printf("%s \n", dp->d_name);
        }

        closedir(dirp);
}
```

2.5 b

```
// Q 2.5B

#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <dirent.h>
#include <string.h>

int main(int argc, char **argv)
{
        DIR *dirp;
        struct dirent *dp;
        char *dir_name;

        if (argc == 1)
                dir_name = ".";
        else
                dir_name = argv[1];

        dirp = opendir(dir_name);

        if (dirp == NULL) {
                fprintf(stderr, "Error: can't open '%s'\n", dir_name);
                exit(1);
        }
```

```
        while ((dp = readdir(dirp)) != NULL) {
                if (strcmp(dp->d_name, ".") == 0 || strcmp(dp->d_name, "..") == 0)
                        continue;
                else
                        printf("%s \n", dp->d_name);
        }

        closedir(dirp);
}

2.5 c
// Q 2.5C

#define _POSIX_SOURCE 1

#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <dirent.h>
#include <string.h>
#include <errno.h>
#include <time.h>

int main(int argc, char **argv)
{
        DIR *dirp;
        struct dirent *dp;
        struct stat *statinfo;
        char *dir_name;

        int long_list = 0;

        if (argc == 1)
                dir_name = ".";
        else if (strcmp(argv[1], "-l")==0){
                dir_name = argv[2];
                long_list = 1;
        } else {
                dir_name = argv[1];
        }

        if (long_list) {
                statinfo = (struct stat *)malloc(sizeof(struct stat));

                if (stat(dir_name, statinfo) != 0)
                        fprintf(stderr, "stat() failed: %s\n", strerror(errno));
                else {
                        printf("%07o\t%s\t%10lu\t%s\n", statinfo->st_mode, dir_name, statinfo->st_size,
ctime(&statinfo->st_atime));
                }

                exit(0);
        }
        else {
                dirp = opendir(dir_name);

                if (dirp == NULL) {
                        fprintf(stderr, "Error: can't open '%s'\n", dir_name);
                        exit(1);
                }

                while ((dp = readdir(dirp)) != NULL) {
                        if (strcmp(dp->d_name, ".") == 0 || strcmp(dp->d_name, "..") == 0)
                                continue;
                        else
```

```
                        printf("%s \n", dp->d_name);
                }

                closedir(dirp);
        }
}
```