



Inspiring Excellence

**LAB ASSIGNMENT: 01**

**SPRING 2024**

**COURSE CODE: CSE321**

**COURSE TITLE: Operating System**

**PREPARED BY**

Name: Nur-E-Jannat

ID: 21301744

Section: 12

Email: [nur.e.jannat@g.bracu.ac.bd](mailto:nur.e.jannat@g.bracu.ac.bd)

## 1.

**a.** touch 21301744\_1.txt 21301744\_2.txt 21301744\_3.txt 21301744\_4.txt

mkdir jannat1 jannat2

cp 21301744\_1.txt 21301744\_2.txt jannat2/

mv 21301744\_1.txt 21301744\_2.txt 21301744\_3.txt jannat1/

mkdir jannat3

cp -r jannat1 jannat3

cd jannat3

ls -l

chmod go=rx jannat1/\*

chmod go=rx jannat1

cd ..

ls -la

mv jannat3/

rm -rf \*

```

ubuntu@ubuntu-VirtualBox:~$ cd Desktop
ubuntu@ubuntu-VirtualBox:~/Desktop$ mkdir cse321_lab_a1
ubuntu@ubuntu-VirtualBox:~/Desktop$ cd cse321_lab_a1
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1$ touch 21301744_1.txt 21301744_2.txt 21301744_3.txt 21301744_4.txt
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1$ mkdir jannat1 jannat2
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1$ cp 21301744_1.txt 21301744_2.txt jannat2/
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1$ mv 21301744_1.txt 21301744_2.txt 21301744_3.txt jannat1/
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1$ mkdir jannat3
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1$ cp -r jannat1 jannat3
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1$ cd jannat3
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1/jannat3$ ls -l
total 4
drwxrwxr-x 2 ubuntu ubuntu 4096 ফেব 29 00:40 jannat1
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1/jannat3$ chmod go=rx jannat1/*
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1/jannat3$ chmod go=rx jannat1
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1/jannat3$ ls -l
total 4
drwxr-xr-x 2 ubuntu ubuntu 4096 ফেব 29 00:40 jannat1
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1/jannat3$ cd ..
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1$ ls -la
total 20
drwxrwxr-x 5 ubuntu ubuntu 4096 ফেব 29 00:39 .
drwxr-xr-x 4 ubuntu ubuntu 4096 ফেব 29 00:37 ..
-rw-rw-r-- 1 ubuntu ubuntu 0 ফেব 29 00:38 21301744_4.txt
drwxrwxr-x 2 ubuntu ubuntu 4096 ফেব 29 00:39 jannat1
drwxrwxr-x 2 ubuntu ubuntu 4096 ফেব 29 00:38 jannat2
drwxrwxr-x 3 ubuntu ubuntu 4096 ফেব 29 00:40 jannat3
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1$ mv jannat3 /
mv: cannot move 'jannat3' to '/jannat3': Permission denied
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1$ cd jannat3
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1/jannat3$ mv jannat3 /
mv: cannot stat 'jannat3': No such file or directory
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1/jannat3$ cd ..
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1$ rm -rf *
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1$

```

**b.** touch courseinfo.txt  
grep -c "CSE" courseinfo.txt

```

ubuntu@ubuntu-VirtualBox: ~/Desktop/cse321_lab_a1
courseinfo.txt
~/Desktop/cse321_lab_a1
1 CSE321
2 Operating system
3 12
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1$ touch courseinfo.txt
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1$ grep -c "CSE" courseinfo.txt
1
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1$

```

**c.** sed -n '5,17p' courseinfo.txt

```

ubuntu@ubuntu-VirtualBox: ~/Desktop/cse321_lab_a1
courseinfo.txt
~/Desktop/cse321_lab_a1
1 Anthropology
2 Culture
3 Anthropologists
4 Human subjects
5 Local environment.
6 customs
7 Ethnography.
8 Societies and cultures
9 Ethnocentrism
10 groups as inferior
11 cultural group.
12 Relativism
13 holistic
14 ethnography
15 diversity
16 nacirema
17 crisis
18 environment
19 ok
20
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1$ sed -n '5,17p' courseinfo.txt
Local environment.
Customs
Ethnography.
Societies and cultures
Ethnocentrism
groups as inferior
cultural group.
Relativism
holistic
ethnography
diversity
nacirema
crisis
ubuntu@ubuntu-VirtualBox:~/Desktop/cse321_lab_a1$

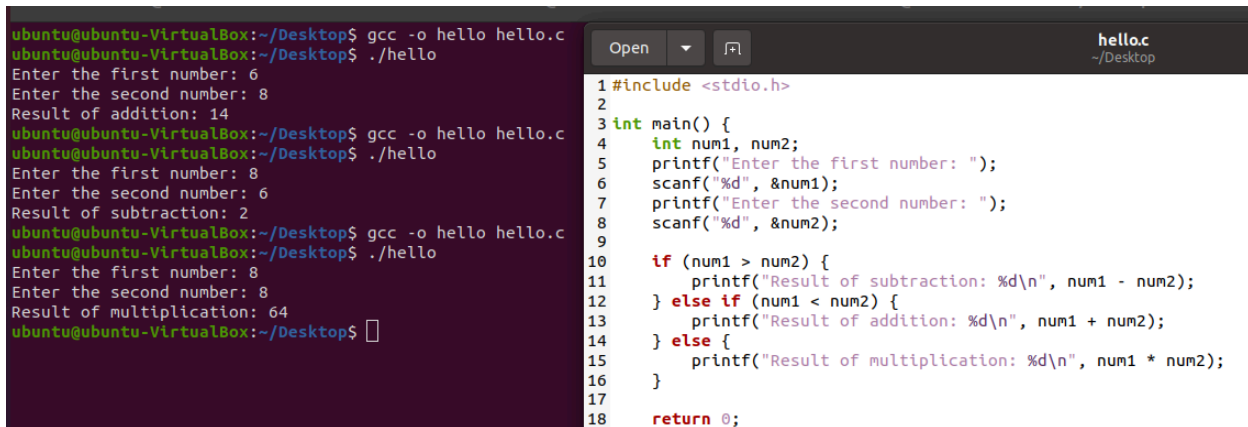
```

## 2.

a.

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

```
int main() {  
    int num1, num2;  
    printf("Enter the first number: ");  
    scanf("%d", &num1);  
    printf("Enter the second number: ");  
    scanf("%d", &num2);  
  
    if (num1 > num2) {  
        printf("Result of subtraction: %d\n", num1 - num2);  
    } else if (num1 < num2) {  
        printf("Result of addition: %d\n", num1 + num2);  
    } else {  
        printf("Result of multiplication: %d\n", num1 * num2);  
    }  
  
    return 0;  
}
```



The screenshot shows a terminal window on the left and a code editor on the right. The terminal window displays the compilation and execution of the C program. The code editor shows the source code of the program, which is a C program that takes two integers as input and performs arithmetic operations based on their comparison.

```
ubuntu@ubuntu-VirtualBox:~/Desktop$ gcc -o hello hello.c  
ubuntu@ubuntu-VirtualBox:~/Desktop$ ./hello  
Enter the first number: 6  
Enter the second number: 8  
Result of addition: 14  
ubuntu@ubuntu-VirtualBox:~/Desktop$ gcc -o hello hello.c  
ubuntu@ubuntu-VirtualBox:~/Desktop$ ./hello  
Enter the first number: 8  
Enter the second number: 6  
Result of subtraction: 2  
ubuntu@ubuntu-VirtualBox:~/Desktop$ gcc -o hello hello.c  
ubuntu@ubuntu-VirtualBox:~/Desktop$ ./hello  
Enter the first number: 8  
Enter the second number: 8  
Result of multiplication: 64  
ubuntu@ubuntu-VirtualBox:~/Desktop$
```

```
1 #include <stdio.h>  
2  
3 int main() {  
4     int num1, num2;  
5     printf("Enter the first number: ");  
6     scanf("%d", &num1);  
7     printf("Enter the second number: ");  
8     scanf("%d", &num2);  
9  
10    if (num1 > num2) {  
11        printf("Result of subtraction: %d\n", num1 - num2);  
12    } else if (num1 < num2) {  
13        printf("Result of addition: %d\n", num1 + num2);  
14    } else {  
15        printf("Result of multiplication: %d\n", num1 * num2);  
16    }  
17  
18    return 0;  
}
```

**b.**

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

void remove_extra_spaces(const char *input_file_name, const char *output_file_name)
{
    FILE *input_file = fopen(input_file_name, "r");
    FILE *output_file = fopen(output_file_name, "w");

    if (input_file == NULL || output_file == NULL) {
        printf("Error.\n");
        return;
    }

    char prev_char = ' ';
    char current_char;

    while ((current_char = fgetc(input_file)) != EOF) {
        if (!isspace(prev_char) || !isspace(current_char)) {
            fputc(current_char, output_file);
        }
        prev_char = current_char;
    }

    fclose(input_file);
    fclose(output_file);

    printf("Extra spaces removed successfully.\n");
}

int main() {
    const char *input_file_name = "input.txt";
    const char *output_file_name = "output.txt";

    remove_extra_spaces(input_file_name, output_file_name);

    return 0;
}
```

```
ubuntu@ubuntu-VirtualBox:~/Desktop$ gcc -o hello hello.c
ubuntu@ubuntu-VirtualBox:~/Desktop$ ./hello
Extra spaces removed successfully.
ubuntu@ubuntu-VirtualBox:~/Desktop$
```

Open  inpu... Save

1 I love Python Programming.

Open  out... Save

1 I love Python Programming.

Plain Text Tab Width: 8 Ln 1, Col 1 INS

Plain Text Tab Width: 8 Ln 1, Col 10 INS

Open  hello.c Save

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <ctype.h>
4
5 void remove_extra_spaces(const char *input_file_name, const char *output_file_name) {
6     FILE *input_file = fopen(input_file_name, "r");
7     FILE *output_file = fopen(output_file_name, "w");
8
9     if (input_file == NULL || output_file == NULL) {
10         printf("Error.\n");
11         return;
12     }
13
14     char prev_char = ' ';
15     char current_char;
16
17     while ((current_char = fgetc(input_file)) != EOF) {
18         if (!isspace(prev_char) || !isspace(current_char)) {
19             fputc(current_char, output_file);
20         }
21         prev_char = current_char;
22     }
23
24     fclose(input_file);
25     fclose(output_file);
26
27     printf("Extra spaces removed successfully.\n");
28 }
29
30 int main() {
31     const char *input_file_name = "input.txt";
32     const char *output_file_name = "output.txt";
33
34     remove_extra_spaces(input_file_name, output_file_name);
35
36     return 0;
37 }
38
```

**C.**

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

```
int is_lowercase(const char *password) {
    for (int i = 0; password[i] != '\0'; i++) {
        if (islower(password[i])) {
            return 1;
        }
    }
    return 0;
}
```

```
int is_uppercase(const char *password) {
    for (int i = 0; password[i] != '\0'; i++) {
        if (isupper(password[i])) {
            return 1;
        }
    }
    return 0;
}
```

```
int is_digit(const char *password) {
    for (int i = 0; password[i] != '\0'; i++) {
        if (isdigit(password[i])) {
            return 1;
        }
    }
    return 0;
}
```

```
int is_special_char(const char *password) {
    const char *special_char = "_,$#@";
    for (int i = 0; password[i] != '\0'; i++) {
        if (strchr(special_char, password[i]) != NULL) {
            return 1;
        }
    }
}
```

```

    }
    return 0;
}

void check_password(const char *password) {
    int lowercase_present = is_lowercase(password);
    int uppercase_present = is_uppercase(password);
    int digit_present = is_digit(password);
    int special_char_present = is_special_char(password);

    if (!lowercase_present) {
        printf("Lowercase character missing\n");
    }
    if (!uppercase_present) {
        printf("Uppercase character missing\n");
    }
    if (!digit_present) {
        printf("Digit missing\n");
    }
    if (!special_char_present) {
        printf("Special character missing\n");
    }

    if (lowercase_present && uppercase_present && digit_present &&
special_char_present) {
        printf("OK\n");
    }
}

int main() {
    char password[100];

    printf("Input: ");
    fgets(password, sizeof(password), stdin);

    if (password[strlen(password) - 1] == '\n') {
        password[strlen(password) - 1] = '\0';
    }

    printf("Output:\n");

```



```

    check_password(password);

    return 0;
}

```

```

ubuntu@ubuntu-VirtualBox:~/Desktop$ gcc -o hello hello.c
ubuntu@ubuntu-VirtualBox:~/Desktop$ ./hello
Input: Jannat#34343
Output:
OK
ubuntu@ubuntu-VirtualBox:~/Desktop$ gcc -o hello hello.c
ubuntu@ubuntu-VirtualBox:~/Desktop$ ./hello
Input: jannat#32
Output:
Uppercase character missing
ubuntu@ubuntu-VirtualBox:~/Desktop$ gcc -o hello hello.c
ubuntu@ubuntu-VirtualBox:~/Desktop$ ./hello
Input: 121348910
Output:
Lowercase character missing
Uppercase character missing
Special character missing
ubuntu@ubuntu-VirtualBox:~/Desktop$

```

```

3 #include <ctype.h>
4
5 int is_lowercase(const char *password) {
6     for (int i = 0; password[i] != '\0'; i++) {
7         if (islower(password[i])) {
8             return 1;
9         }
10    }
11    return 0;
12 }
13
14 int is_uppercase(const char *password) {
15     for (int i = 0; password[i] != '\0'; i++) {
16         if (isupper(password[i])) {
17             return 1;
18         }
19    }
20    return 0;
21 }
22
23 int is_digit(const char *password) {
24     for (int i = 0; password[i] != '\0'; i++) {
25         if (isdigit(password[i])) {
26             return 1;
27         }
28    }
29    return 0;
30 }
31
32 int is_special_char(const char *password) {
33     const char *special_char = "_,$#@";
34     for (int i = 0; password[i] != '\0'; i++) {
35         if (strchr(special_char, password[i]) != NULL) {
36             return 1;
37         }
38    }
39    return 0;
40 }
41
42 void check_password(const char *password) {
43     int lowercase_present = is_lowercase(password);
44     int uppercase_present = is_uppercase(password);
45     int digit_present = is_digit(password);
46     int special_char_present = is_special_char(password);
47

```

d.

```

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

```

```

int is_updated_email(const char *email) {
    const char *old_domain = "@kaaj.com";
    const char *new_domain = "@sheba.xyz";

```

```

    const char *at_symbol = strchr(email, '@');

```

```
if (at_symbol == NULL) {  
    return 0;  
}
```

```
if (strcmp(at_symbol, old_domain) == 0) {  
    return 0;  
} else if (strcmp(at_symbol, new_domain) == 0) {  
    return 1;  
} else {  
    return 0;  
}  
}
```

```
int main() {  
    char email[100];  
  
    printf("Enter the email address: ");  
    scanf("%s", email);  
  
    if (is_updated_email(email)) {  
        printf("Email address is okay\n");  
    } else {  
        printf("Email address is outdated\n");  
    }  
  
    return 0;  
}
```

```

ubuntu@ubuntu-VirtualBox:~/Desktop$ gcc -o hello hello.c
ubuntu@ubuntu-VirtualBox:~/Desktop$ ./hello
Enter the email address: fahmid@kaaaj.com
Email address is outdated
ubuntu@ubuntu-VirtualBox:~/Desktop$ gcc -o hello hello.c
ubuntu@ubuntu-VirtualBox:~/Desktop$ ./hello
Enter the email address: zaki@sheba.xyz
Email address is okay
ubuntu@ubuntu-VirtualBox:~/Desktop$ 

```

```

1 #include <stdio.h>
2 #include <string.h>
3
4 int is_updated_email(const char *email) {
5     const char *old_domain = "@kaaaj.com";
6     const char *new_domain = "@sheba.xyz";
7
8
9     const char *at_symbol = strchr(email, '@');
10    if (at_symbol == NULL) {
11        return 0;
12    }
13
14
15    if (strcmp(at_symbol, old_domain) == 0) {
16        return 0;
17    } else if (strcmp(at_symbol, new_domain) == 0) {
18        return 1;
19    } else {
20        return 0;
21    }
22 }
23
24 int main() {
25     char email[100];
26
27     printf("Enter the email address: ");
28     scanf("%s", email);
29
30     if (is_updated_email(email)) {
31         printf("Email address is okay\n");
32     } else {
33         printf("Email address is outdated\n");
34     }
35
36     return 0;
37 }
38

```

e.

```

#include <stdio.h>
#include <string.h>
#include <ctype.h>

```

```

int is_palindrome(const char *str) {
    const char *start = str;
    const char *end = str + strlen(str) - 1;

    while (start < end) {
        while (!isalnum(*start) && start < end) {
            start++;
        }
        while (!isalnum(*end) && start < end) {
            end--;
        }

        if (tolower(*start) != tolower(*end)) {

```

```
        return 0;
    }

    start++;
    end--;
}

return 1;
}

int main() {
    char input[100];

    printf("Enter a string: ");
    fgets(input, sizeof(input), stdin);
    if (input[strlen(input) - 1] == '\n') {
        input[strlen(input) - 1] = '\0';
    }

    if (is_palindrome(input)) {
        printf("Palindrome\n");
    } else {
        printf("Not Palindrome\n");
    }

    return 0;
}
```

```
ubuntu@ubuntu-VirtualBox:~/Desktop$ gcc -o hello hello.c
ubuntu@ubuntu-VirtualBox:~/Desktop$ ./hello
Enter a string: 6778uhu
Not Palindrome
ubuntu@ubuntu-VirtualBox:~/Desktop$ gcc -o hello hello.c
ubuntu@ubuntu-VirtualBox:~/Desktop$ ./hello
Enter a string: abccba
Palindrome
ubuntu@ubuntu-VirtualBox:~/Desktop$
```

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <ctype.h>
4
5 int is_palindrome(const char *str) {
6     const char *start = str;
7     const char *end = str + strlen(str) - 1;
8
9     while (start < end) {
10         while (!isalnum(*start) && start < end) {
11             start++;
12         }
13         while (!isalnum(*end) && start < end) {
14             end--;
15         }
16
17         if (tolower(*start) != tolower(*end)) {
18             return 0;
19         }
20
21         start++;
22         end--;
23     }
24
25     return 1;
26 }
27
28 int main() {
29     char input[100];
30
31     printf("Enter a string: ");
32     fgets(input, sizeof(input), stdin);
33     if (input[strlen(input) - 1] == '\n') {
34         input[strlen(input) - 1] = '\0';
35     }
36
37     if (is_palindrome(input)) {
38         printf("Palindrome\n");
39     } else {
40         printf("Not Palindrome\n");
41     }
42
43     return 0;
44 }
45
```