Assignment 2 - Processes and Threads

1) What is the significance of Environmental Variables in Linux ?

a) Write a C program to print the following environmental variables:

i) USER
ii) HOME
iii) PWD
iv) PATH

b) Modify the above C program to also add a new environmental variable named "ROOT" with a value "root". If the variable already exist, overwrite its value to the above mentioned one.

2) Mutexes and Threads

a) Write a program which spawns a thread and uses the thread to output a simple "Hello World" to the screen.

b) Write a program which
i)spawns a thread,
ii)accepts a value from the user using the main program thread,
iii)passes the value to the newly spawned thread through a shared variable, and
iv)uses the newly spawned thread to output the value back to the user
Ensure synchronisation and eliminate possibilities of errors such as in "Producer-Consumer" problem by using synchronization primitives wherever needed.

3)

a)Execute the given program and depending on the output explain the sequence of execution of the program.

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

static void print_os();

pthread_mutex_t mutex;

void* print_xs(void* unused) {
    int j;
    pthread_mutex_lock(&mutex);
    for(j = 0; j < 1000; j++)
        fputc(’x’, stderr);
    pthread_mutex_unlock(&mutex);
    return NULL;
}

static void print_os() {
    int i;

    for(i = 0; i < 1000; i++)
        fputc(’o’, stderr);

}

int main() {
    pthread_t new;
    pthread_create(&new, NULL, &print_xs, NULL);
    pthread_mutex_lock(&mutex);
    print_os();
    pthread_mutex_unlock(&mutex);
    pthread_join(new, NULL);
    return 0;
}
```

b) Modify the given program such that the function print_xs executes before the function print_os using a synchronization primitive.

4) Write a program that
a) spawns a thread,
b) accepts two values from the user using the main program thread,
c) passes both values to the newly created thread through a shared variable,
d) uses the newly created thread to carry out multiplication operation upon the aforementioned values,
e) communicates the result of communication from the newly created thread back to the main program thread again through the same shared variable
f) and finally uses the main program thread to display the result back to the user.
Also make sure that everything is properly synchronised and error-free by the usage of synchronization primitives.