

Name: Abrar Khatib Lajim

ID:2022-3-60-043

Section:01

CSE 325: LAB 01 (Exercise - Offline)

Course Instructor: Khairum Islam Seme

1. Create a program that creates a child process. The child process prints "I am a child process" 100 times in a loop. Whereas the parent process prints "I am a parent process" 100 times in a loop.

Output:

```
hasanpc@hasanpc-VirtualBox:~/Desktop$ gedit lex1.c
hasanpc@hasanpc-VirtualBox:~/Desktop$ cat lex1.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

int main() {
    pid_t pid = fork();

    if (pid < 0)
    {
        perror("Fork failed");
        return 1;
    }
    else if (pid == 0)
    {
        for (int i = 0; i < 100; i++)
        {
            printf("I am a child process\n");
        }
    }
    else
    {
        for (int i = 0; i < 100; i++)
        {
            printf("I am a parent process\n");
        }

        wait(NULL);
    }
}

return 0;
}
hasanpc@hasanpc-VirtualBox:~/Desktop$
```

This file "/home/hasanpc/Desktop/lex1.c" is already open in another window.
Do you want to edit it anyway?

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

int main()
{
    pid_t pid = fork();

    if (pid < 0)
    {
        perror("Fork failed");
        return 1;
    }
    else if (pid == 0)
    {
        for (int i = 0; i < 100; i++)
        {
            printf("I am a child process\n");
        }
    }
    else
    {
        for (int i = 0; i < 100; i++)
        {
            printf("I am a parent process\n");
        }
        wait(NULL);
    }

    return 0;
}
```

```
File Edit View Search Terminal Help
I am a child process
hasanpc@hasanpc-VirtualBox:~/Desktop$ gedit lex1.c
hasanpc@hasanpc-VirtualBox:~/Desktop$ cat lex1.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

int main() {
    pid_t pid = fork();

    if (pid < 0)
    {
        perror("Fork failed");
        return 1;
    }
    else if (pid == 0)
    {
        for (int i = 0; i < 100; i++)
        {
            printf("I am a child process\n");
        }
    }
    else
    {
        for (int i = 0; i < 100; i++)
        {
            printf("I am a parent process\n");
        }

        wait(NULL);
    }
}

return 0;
}
hasanpc@hasanpc-VirtualBox:~/Desktop$
```

2. Create a program named stat that takes an integer array as an input(delimited by some character such as S). The program then creates 3 child processes each of which does exactly one task from the following.

- a) Adds them and print the result on the screen. (done by child 1)
- b) Shows the average on the screen. (done by child 2)
- c) Print the maximum number on the screen. (done by child 3)

Output:

The screenshot shows a Linux desktop environment with two terminal windows and a file manager. The left terminal window contains the source code for the lab2.c program. The right terminal window contains the implementation of the stat program, which uses fork() to create three child processes for addition, average, and maximum calculations.

```
Running - Oracle VM VirtualBox
File Edit View Search Terminal Help
hasanpc@hasanpc-VirtualBox:~/Desktop$ cat lab2.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>

void add(int *arr, int size) {
    int sum = 0;
    for (int i = 0; i < size; i++) {
        sum += arr[i];
    }
    printf("Sum: %d\n", sum);
}

void average(int *arr, int size) {
    if (size == 0) {
        printf("Average: 0\n");
        return;
    }
    int sum = 0;
    for (int i = 0; i < size; i++) {
        sum += arr[i];
    }
    printf("Average: %.2f\n", (float)sum / size);
}

void maximum(int *arr, int size) {
    if (size == 0) {
        printf("Maximum: 0\n");
        return;
    }
    int max = arr[0];
    for (int i = 1; i < size; i++) {
        if (arr[i] > max) {
            max = arr[i];
        }
    }
    printf("Maximum: %d\n", max);
}

int main(int argc, char *argv[]) {
    if (argc != 2) {
        fprintf(stderr, "Usage: %s <numbers delimited by $>\n", argv[0]);
        exit(1);
    }
    ...

hasanpc@hasanpc-VirtualBox:~/Desktop$ ./stat 1$2$4$5$
bash: ./stat: No such file or directory
hasanpc@hasanpc-VirtualBox:~/Desktop$ ./lab2 1$2$3$4$5$
Sum: 1
Average: 1.00
Maximum: 1
hasanpc@hasanpc-VirtualBox:~/Desktop$
```

```
Ubuntu18 [Running] - Oracle VM VirtualBox
File Edit View Search Terminal Help
fprintf(stderr, "Usage: %s <numbers delimited by $>\n", argv[0]);
return 1;
}

char *input = argv[1];
char *token;
int arr[100];
int size = 0;

token = strtok(input, "$");
while (token != NULL) {
    arr[size++] = atoi(token);
    token = strtok(NULL, "$");
}

pid_t pid1, pid2, pid3;

if ((pid1 = fork()) == 0) {
    add(arr, size);
    exit(0);
} else {
    waitpid(pid1, NULL, 0);
}

if ((pid2 = fork()) == 0) {
    average(arr, size);
    exit(0);
} else {
    waitpid(pid2, NULL, 0);
}

if ((pid3 = fork()) == 0) {
    maximum(arr, size);
    exit(0);
} else {
    waitpid(pid3, NULL, 0);
}

return 0;
}

hasanpc@hasanpc-VirtualBox:~/Desktop$
```

The screenshot shows a terminal window where the lab2.c program is being run. It first attempts to run cddesktop, which fails because it's a command not found. Then it runs cd Desktop, touch lab2.c, gedit lab2.c, gcc lab2.c -o lab2, and ./lab2. The ./lab2 command outputs usage information and then runs with the input 1\$2\$3\$4\$5\$. It prints the sum (1), average (1.00), and maximum (1).

```
File Edit View Search Terminal Help
hasanpc@hasanpc-VirtualBox:~$ cddesktop
cddesktop: command not found
hasanpc@hasanpc-VirtualBox:~$ cd Desktop
hasanpc@hasanpc-VirtualBox:~/Desktop$ touch lab2.c
hasanpc@hasanpc-VirtualBox:~/Desktop$ gedit lab2.c
hasanpc@hasanpc-VirtualBox:~/Desktop$ gcc lab2.c -o lab2
hasanpc@hasanpc-VirtualBox:~/Desktop$ ./lab2
Usage: ./lab2 <numbers delimited by $>
hasanpc@hasanpc-VirtualBox:~/Desktop$ ^C
hasanpc@hasanpc-VirtualBox:~/Desktop$ gedit lab2.c
hasanpc@hasanpc-VirtualBox:~/Desktop$ ^C
hasanpc@hasanpc-VirtualBox:~/Desktop$ gcc lab2.c -o lab2
hasanpc@hasanpc-VirtualBox:~/Desktop$ ./lab2
Usage: ./lab2 <numbers delimited by $>
hasanpc@hasanpc-VirtualBox:~/Desktop$ ./stat 1$2$4$5$
bash: ./stat: No such file or directory
hasanpc@hasanpc-VirtualBox:~/Desktop$ ./lab2 1$2$3$4$5$
Sum: 1
Average: 1.00
Maximum: 1
hasanpc@hasanpc-VirtualBox:~/Desktop$
```