OS Unit 3 & 4 – Assignment

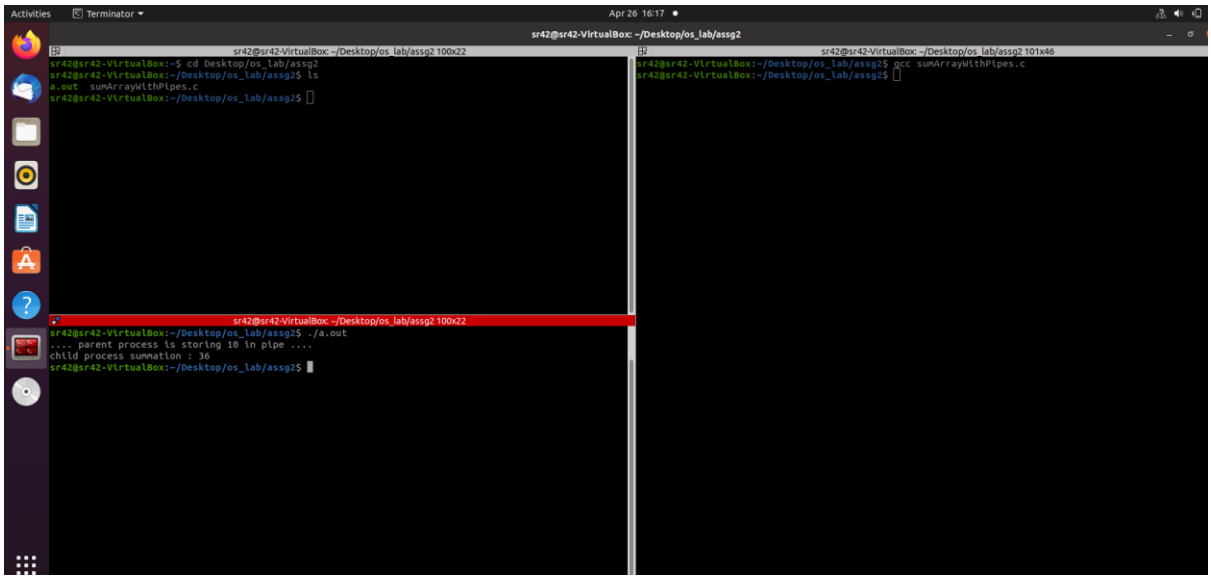
Sriram R PES1UG20CS435 Section : H Roll # : 16

Source codes & outputs :

1. Finding the sum of an array using pipes.

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

#define ARRAY_SIZE 8
#define BUFFER_SIZE 8 // should be greater than expected number of digits in sum
#define READ_END 0
#define WRITE_END 1

int main()
{
	int numArray[] = { 1, 2, 3, 4, 5, 6, 7, 8 };
	int sumChild = 0;
	int sum = 0;

	char write_msg[BUFFER_SIZE] = "";
	char read_msg[BUFFER_SIZE] = "";
	int fd[2];
	pid_t pid;

	if (pipe(fd) == -1)
	{
		printf("pipe failure\n");
		return 1;
	}

	pid = fork();

	if (pid > 0) // parent process - calculates sum of first half and stores in
pipe
	{
		for (int i = 0; i < (ARRAY_SIZE / 2); i++) sum += numArray[i];
		snprintf(write_msg, BUFFER_SIZE, "%d", sum); // converts integer to
string (itoa not supported)
		printf(".... parent process is storing %s in pipe ....\n", write_msg);

		close(fd[READ_END]);
		write(fd[WRITE_END], write_msg, strlen(write_msg) + 1);
		close(fd[WRITE_END]);
		wait(NULL);
	}

	else if (pid == 0) // child process - calculates sum of second half, adds to
the sum in the pipe and prints
	{
		close(fd[WRITE_END]);
		read(fd[READ_END], read_msg, BUFFER_SIZE);
```

```
                for (int i = (ARRAY_SIZE / 2); i < ARRAY_SIZE; i++) sumChild +=
numArray[i];
                sumChild += atoi(read_msg);

                printf("child process summation : %d\n", sumChild);
                close(fd[READ_END]);
        }

        else
        {
                printf("fork failure\n");
                return 1;
        }

        return 0;
}
```

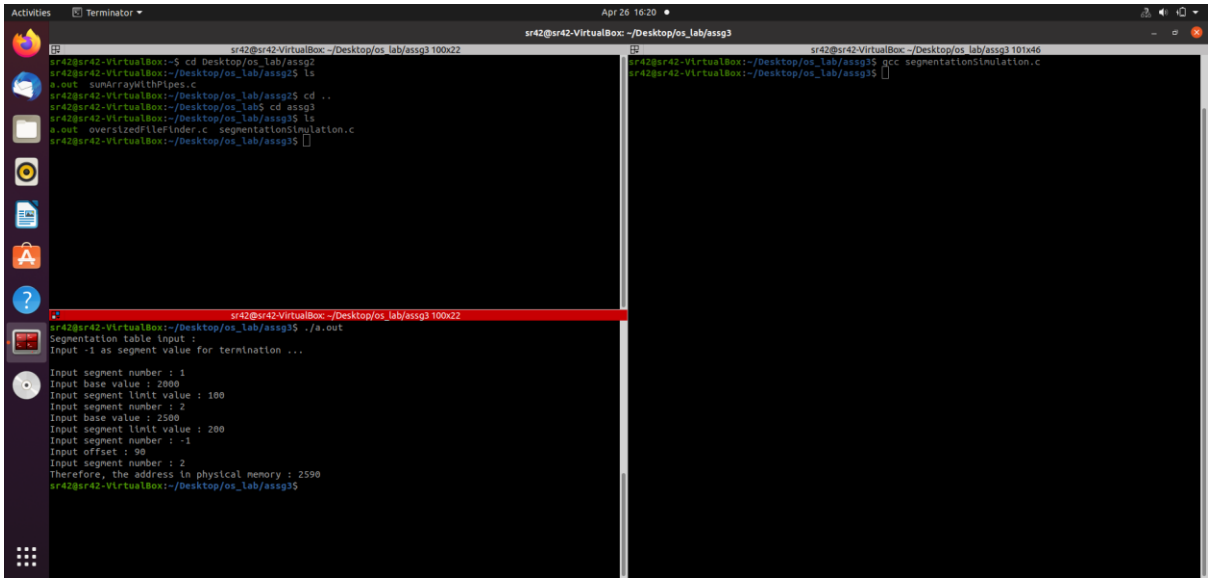


2. Simulation of segmentation on C.

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

struct list
{
        int seg;
        int base;
        int limit;
        struct list* next;
}*p;

void insert(struct list* q, int base, int limit, int seg)
{
        if (p == NULL)
        {
                p = malloc(sizeof(struct list));
                p->limit = limit;
                p->base = base;
                p->seg = seg;
                p->next = NULL;
```

```
        }

        else
        {
                while (q->next != NULL) q = q->next;

                q->next = malloc(sizeof(struct list));
                q->next->limit = limit;
                q->next->base = base;
                q->next->seg = seg;
                q->next->next = NULL;
        }
}

int find(struct list* q, int seg)
{
        while (q->seg != seg) q = q->next;
        return q->limit;
}

int search(struct list* q, int seg)
{
        while (q->seg != seg) q = q->next;
        return q->base;
}

int main()
{
        p = NULL;
        int seg, offset, limit, base, c, s, physical;

        printf("Segmentation table input : \n");
        printf("Input -1 as segment value for termination ...\n\n");

        do
        {
                printf("Input segment number : ");
                scanf("%d", &seg);

                if (seg != -1)
                {
                        printf("Input base value : ");
                        scanf("%d", &base);

                        printf("Input segment limit value : ");
                        scanf("%d", &limit);

                        insert(p, base, limit, seg);
                }
        } while (seg != -1);

        printf("Input offset : ");
        scanf("%d", &offset);

        printf("Input segment number : ");
        scanf("%d", &seg);
        c = find(p, seg);
        s = search(p, seg);

        if (offset < c)
        {
                physical = s + offset;
```

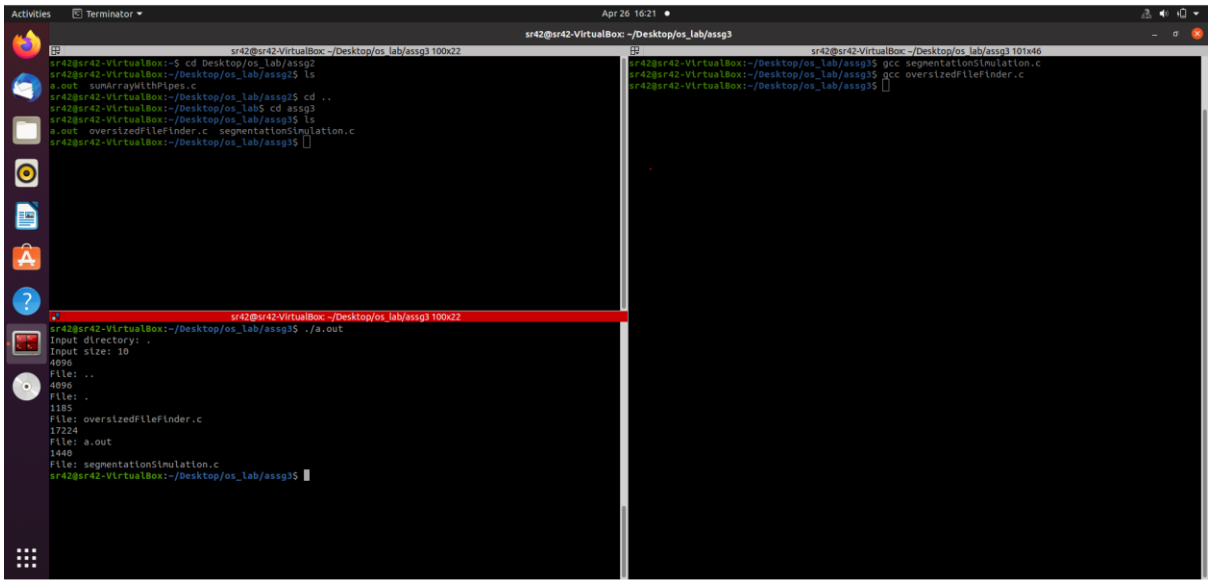
```
                printf("Therefore, the address in physical memory : %d\n", physical);
        }

        else printf("error");

        return 0;
}
```

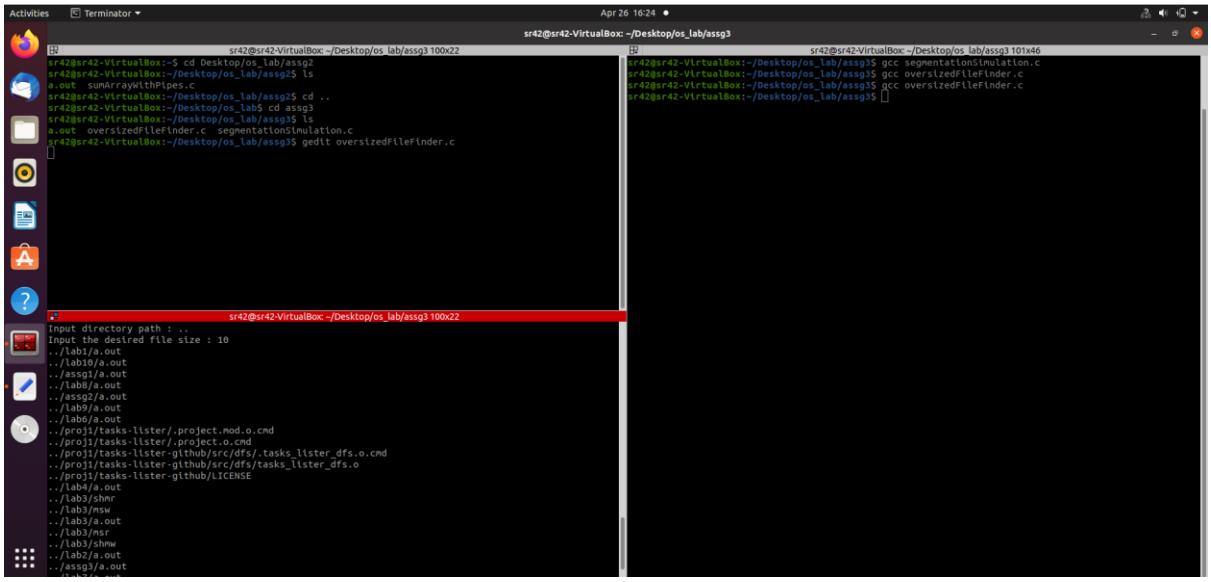


3. Finding all the oversized files in a directory.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <dirent.h>
#include <sys/stat.h>
#include <time.h>

int main()
{
        DIR* d;
        struct dirent* dir;
        struct stat stats;
        long dt;

        char dirname[256];
        printf("Input directory: ");
        fflush(stdout);
        scanf("%s", dirname);
        d = opendir(dirname);

        int size;
        printf("Input size: ");
        scanf("%d", &size);

        if (!d)
        {
                printf("error\n");
```

```
                exit(0);
        }

        while ((dir = readdir(d)))
        {
                if (stat(dir->d_name, &stats) == 0)
                {
                        dt = stats.st_size;
                        printf("%ld\n", dt);

                        if (dt > size) printf("File: %s\n", dir->d_name);
                }
                else printf("Couldn't read file properties of file: %s\n ", dir-
>d_name);
        }
        closedir(d);

        return 0;
}

/*

// alternate method – this one looks WAYY better but apparently using system() wasn't
allowed. I included it's output in the second screenshot anyways

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

int main()
{
        char *command = (char*)malloc(256 * sizeof(char));
        char *path = (char*)malloc(256 * sizeof(char));
        int size;

        printf("Input directory path : ");
        scanf("%s", path);

        printf("Input the desired file size : ");
        scanf("%d", &size);

        sprintf(command, "find %s -size +%dk", path, size);
        system(command);

        return 0;
}

*/
```


Activities
Terminator
Apr 26 16:21
sr42@sr42-VirtualBox: ~/Desktop/os_lab/assg3
sr42@sr42-VirtualBox:~$ cd Desktop/os_lab/assg2
sr42@sr42-VirtualBox:~/Desktop/os_lab/assg2$ ls
a.out sumArrayWithPipes.c
sr42@sr42-VirtualBox:~/Desktop/os_lab/assg2$ cd ..
sr42@sr42-VirtualBox:~/Desktop/os_lab$ cd assg3
sr42@sr42-VirtualBox:~/Desktop/os_lab/assg3$ ls
a.out oversizedFileFinder.c segmentationSimulation.c
sr42@sr42-VirtualBox:~/Desktop/os_lab/assg3$
sr42@sr42-VirtualBox:~/Desktop/os_lab/assg3$ gcc segmentationSimulation.c
sr42@sr42-VirtualBox:~/Desktop/os_lab/assg3$ gcc oversizedFileFinder.c
sr42@sr42-VirtualBox:~/Desktop/os_lab/assg3$
sr42@sr42-VirtualBox:~/Desktop/os_lab/assg3$ ./a.out
Input directory: .
Input size: 10
4096
File: ..
4096
File: .
1185
File: oversizedFileFinder.c
17224
File: a.out
1448
File: segmentationSimulation.c
sr42@sr42-VirtualBox:~/Desktop/os_lab/assg3$


Activities
Terminator
Apr 26 16:24
sr42@sr42-VirtualBox: ~/Desktop/os_lab/assg3
sr42@sr42-VirtualBox:~$ cd Desktop/os_lab/assg2
sr42@sr42-VirtualBox:~/Desktop/os_lab/assg2$ ls
a.out sumArrayWithPipes.c
sr42@sr42-VirtualBox:~/Desktop/os_lab/assg2$ cd ..
sr42@sr42-VirtualBox:~/Desktop/os_lab$ cd assg3
sr42@sr42-VirtualBox:~/Desktop/os_lab/assg3$ ls
a.out oversizedFileFinder.c segmentationSimulation.c
sr42@sr42-VirtualBox:~/Desktop/os_lab/assg3$ gedit oversizedFileFinder.c
sr42@sr42-VirtualBox:~/Desktop/os_lab/assg3$ gcc segmentationSimulation.c
sr42@sr42-VirtualBox:~/Desktop/os_lab/assg3$ gcc oversizedFileFinder.c
sr42@sr42-VirtualBox:~/Desktop/os_lab/assg3$ gcc oversizedFileFinder.c
sr42@sr42-VirtualBox:~/Desktop/os_lab/assg3$
Input directory path : ..
Input the desired file size : 10
../lab1/a.out
../lab10/a.out
../assg1/a.out
../lab8/a.out
../assg2/a.out
../lab9/a.out
../lab6/a.out
../proj1/tasks-lister/.project.mod.o.cmd
../proj1/tasks-lister/.project.o.cmd
../proj1/tasks-lister-github/src/dfs/.tasks_lister_dfs.o.cmd
../proj1/tasks-lister-github/src/dfs/tasks_lister_dfs.o
../proj1/tasks-lister-github/LICENSE
../lab4/a.out
../lab3/shmr
../lab3/msw
../lab3/a.out
../lab3/msr
../lab3/shmw
../lab2/a.out
../assg3/a.out