

**CSE344
SYSTEM
PROGRAMMING
COURSE
HW2**

**171044034
Mustafa
Gurler**

Design

I read file to a buffer. I saved them on a three dimensional pointer and send them one by one to child process.

```
close(fd);
pid_t handler_signal[r_ith] ;
char *programName = "./childProcess";
int status;
pid_t childPid;
for(int i=0 ; i<r_ith ; i++){
    childPid = fork();

    if(childPid > 0){
        handler_signal[i] = childPid;
    }
    if(childPid == -1){
        perror("fork");
        exit(1);
    }else if(childPid == 0){

        execve(programName, argv, dimensional_coordinateates[i]);
        exit(1);
    }
}

while ((childPid = wait(&status)) > 0){

    if(signal_arrived == 1){
        for(int i=0 ; i<r_ith ; i++){
            kill(handler_signal[i] , SIGINT);
        }

        exit(1);
    }

    if(childPid == -1){
        if(errno == ECHILD){
            break;
        }else{

```

I find all the possible input for children and saved them in r_ith;

Signals was also created on main process and I forwarded them to children processes.

After I created my child I calculate convariance matrix one by one. By the mean time, Main process was waiting all the children die.

```
while ((childPid = wait(&status)) > 0){

    if(signal_arrived == 1){
        for(int i=0 ; i<r_ith ; i++){
            kill(handler_signal[i] , SIGINT);
        }

        exit(1);
    }

    if(childPid == -1){
        if(errno == ECHILD){
            break;
        }else{
            exit(EXIT_FAILURE);
        }
    }
}
```

After I collect all my calculations from children processes. I also calculate Frobenius Forms.

```
free(norms);

double normss[r_ith];
for(int i=0 ; i<r_ith-1 ; i++){
    normss[i] = frobeniusNorm(norm_double[i]);
}

double min = 100000000;
int first = -1;
int second = -1;
for(int i=0 ; i < r_ith ; i++){
    for(int j=0 ; j < r_ith ; j++){
        if(fabs(normss[i]- normss[j]) <= min ){
            if(i != j){
                min = fabs(normss[i] - normss[j]);
                first = i;
                second = j;
            }
        }
    }
}
```

I also free all the possible memory leaks and Valgrind shows me no memory leak.

Expected Result shown in the below:

[illegible]