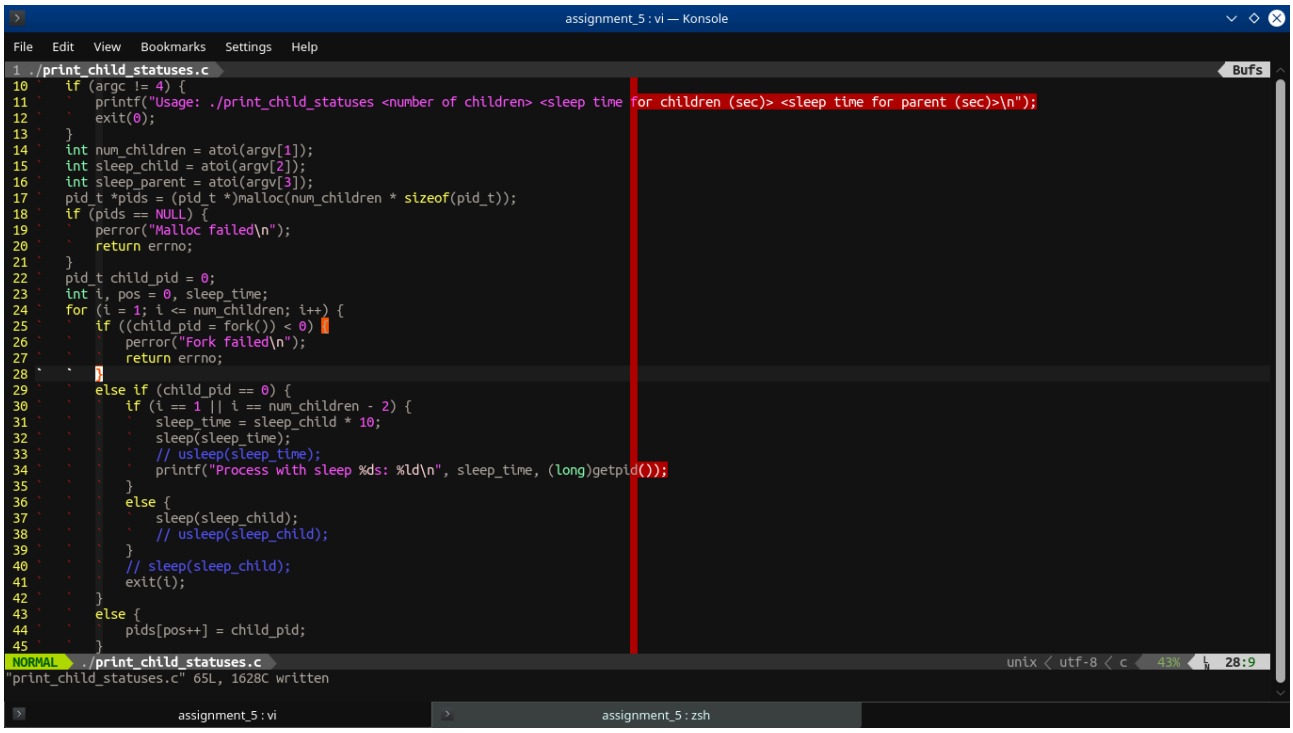
Assignment 5

Niramay Vaidya 111605075
Srishti Shelke 111603056

1. The parent starts as many child processes as to the value of its integer command line argument. The child processes simply sleep for the time specified by the argument, then exit. After starting all the children, the parent process does not wait for them immediately, but after a time specified by command line argument, checks the status of all terminated children, print the list of non terminated children and then terminates itself.

The structure of the code is such that it is next to impossible to create a scenario when part of the forked children have terminated the rest have not. This is because the code contains a for loop in the parent which sequentially forks the number of children as specified by the command line argument and then prints out the termination status of these children sequentially in the same order using a for loop again. Since the sleep time for all children is same, it is highly unlikely that the above scenario will occur irrespective of using sleep/usleep (keeping sleep time in secs or usecs) and if the first child has terminated then all following children have also terminated and if it hasn't terminated, then all following children have also not terminated yet. In order to demonstrate the above scenario, the code makes a provision of changing the sleep time of some of the children specifically to a value with one degree greater than the sleep time passed as the command line argument for other children. This was, these specific children have not yet terminated while others have as seen in the second output screenshot in the actual submission.

Code-




```c
    if (argc != 4) {
        printf("Usage: ./print_child_statuses <number of children> <sleep time for children (sec)> <sleep time for parent (sec)>\n");
        exit(0);
    }
    int num_children = atoi(argv[1]);
    int sleep_child = atoi(argv[2]);
    int sleep_parent = atoi(argv[3]);
    pid_t *pids = (pid_t *)malloc(num_children * sizeof(pid_t));
    if (pids == NULL) {
        perror("Malloc failed\n");
        return errno;
    }
    pid_t child_pid = 0;
    int i, pos = 0, sleep_time;
    for (i = 1; i <= num_children; i++) {
        if ((child_pid = fork()) < 0) {
            perror("Fork failed\n");
            return errno;
        }
        else if (child_pid == 0) {
            if (i == 1 || i == num_children - 2) {
                sleep_time = sleep_child * 10;
                sleep(sleep_time);
                // usleep(sleep_time);
                printf("Process with sleep %ds: %ld\n", sleep_time, (long)getpid());
            }
            else {
                sleep(sleep_child);
                // usleep(sleep_child);
            }
            // sleep(sleep_child);
            exit(i);
        }
        else {
            pids[pos++] = child_pid;
        }
```

Output-

```
niramay@niramay  ~/Documents/sem7/AUP/assignment_5  ./print_child_statuses 25 1 2                    ✔  10010  19:25:23
Process with pid 4897 not yet terminated
Process with pid 4898 terminated with status 2
Process with pid 4899 terminated with status 3
Process with pid 4900 terminated with status 4
Process with pid 4901 terminated with status 5
Process with pid 4902 terminated with status 6
Process with pid 4903 terminated with status 7
Process with pid 4904 terminated with status 8
Process with pid 4905 terminated with status 9
Process with pid 4906 terminated with status 10
Process with pid 4907 terminated with status 11
Process with pid 4908 terminated with status 12
Process with pid 4909 terminated with status 13
Process with pid 4910 terminated with status 14
Process with pid 4911 terminated with status 15
Process with pid 4912 terminated with status 16
Process with pid 4913 terminated with status 17
Process with pid 4914 terminated with status 18
Process with pid 4915 terminated with status 19
Process with pid 4916 terminated with status 20
Process with pid 4917 terminated with status 21
Process with pid 4918 terminated with status 22
Process with pid 4919 not yet terminated
Process with pid 4920 terminated with status 24
Process with pid 4921 terminated with status 25
niramay@niramay  ~/Documents/sem7/AUP/assignment_5  Process with sleep 10s: 4897                    ✔  10011  19:25:30
Process with sleep 10s: 4919

niramay@niramay  ~/Documents/sem7/AUP/assignment_5                                                   1 ✘  10011  19:26:30
```