

Niraj Bhatta

Dr. Patrick McDowell

CMPS_431

Partial Batch Simulation

Homework 0

This assignment involves writing code for a partial batch simulation in C, where a program will compile and run following specific instructions. The batch simulation consists of a batch folder containing the main file batchmonitor.c, which manages the execution of jobs. Inside the jobs folder, small C programs are stored, which are compiled and executed by the batch monitor to simulate batch processing."

The root folder Structure:

```
This message is shown once a day. To disable it please create the
/home/niraj0010/.hushlogin file.
niraj0010@LAPTOP-VL2JH4U5:~$ cd /mnt/c/Users/neera/batch
niraj0010@LAPTOP-VL2JH4U5:/mnt/c/Users/neera/batch$ tree
.
├── Extra
│   ├── pattern.c
│   └── pattern.exe
├── batchmonitor
├── batchmonitor.c
└── batchmonitor.exe
jobs
├── Factorial.c
├── Factorial.exe
├── Factorial.out
├── Summation.c
├── Summation.exe
├── Summation.out
├── dots.c
├── dots.exe
├── dots.out
├── hash.c
├── hash.exe
└── hash.out

2 directories, 17 files
niraj0010@LAPTOP-VL2JH4U5:/mnt/c/Users/neera/batch$
```

The small jobs like printing dots, hashes, summation and factorial are inside jobs folder. The code for those jobs are:

Dots.c

```
C Summation.c | C batchmonitor.c | C dots.c
jobs > C dots.c > ...
1 #include <stdio.h>
2 int main() {
3
4     for (int i = 0; i < 50; i++) {
5         printf(".");
6     }
7
8     printf("\n");
9
10    return 0;
11 }
12 }
```

Hash.c

```
#include <stdio.h>

int main()
{
    for (int i = 0; i < 100; i++)
    {
        printf("#");
    }

    printf("\n");
}

return 0;
```

Summation.c

```
jobs > C Summation.c > ⌂ main()
1 #include <stdio.h>
2
3 int main() {
4     int sum = 0;
5
6
7     for (int i = 1; i <= 10; i++) {
8         sum += i;
9         printf("Sum after adding %d: %d\n", i, sum);
10    }
11
12    return 0;
13 }
14 }
```

Factorial.c

```
1 #include <stdio.h>
2
3 int main() {
4     int factorial = 1;
5
6
7     for (int i = 1; i <= 10; i++) {
8         factorial *= i;
9         printf("Factorial after multiplying by %d: %d\n", i, factorial);
10    }
}
```

There is another folder that lies inside batch but is different from the jobs folder. The job called pattern.c is inside the folder named “Extra”

Pattern.c

```
1 #include <stdio.h>
2
3 int main() {
4     int rows, i, j;
5
6
7     printf("Enter the number of rows: ");
8     scanf("%d", &rows);
9
10
11    for(i = 1; i <= rows; i++) {
12
13        for(j = 1; j <= i; j++) {
14            printf("*");
15        }
16
17        printf("\n");
18    }
19
20    return 0;
21 }
22 }
```

The code for Batch Simulator:

```
[batchmonitor.c] ⌂ listJobs(char *)
1 //Niraj Bhatta
2 //W0763241
3 //OS_Program_0_BatchSimulator
4 #include <stdio.h>
5 #include <stdlib.h>
6 #include <dirent.h>
7 #include <string.h>
8
9
10 void runAllTasks(char *directoryPath) {
11     DIR *dirPointer;
12     struct dirent *entry;
13     dirPointer = opendir(directoryPath);
14     if (dirPointer) {
15         while ((entry = readdir(dirPointer)) != NULL) {
16             char *fileName = entry->d_name;
17
18             if (strstr(fileName, ".c") != NULL) {
19                 printf("Compiling and Running Job: %s\n", fileName);
20
21                 char outputName[58];
22                 strncpy(outputName, fileName, strlen(fileName) - 2);
23                 outputName[strlen(fileName) - 2] = '\0';
24
25                 char compileCommand[200];
26                 snprintf(compileCommand, sizeof(compileCommand), "gcc -o \"%s/%s.out\" \"%s/%s\"", directoryPath, outputName, directoryPath, fileName);
27                 system(compileCommand);
28
29                 char runCommand[200];
30                 snprintf(runCommand, sizeof(runCommand), "\"%s/%s.out\"", directoryPath, outputName);
31                 system(runCommand);
32             }
33         }
34         closedir(dirPointer);
35     } else {
36         printf("Could not open directory: %s\n", directoryPath);
37     }
38 }
39
40
```

```
41 void ListJobs(char *directoryPath) {
42     DIR *dirPointer;
43     struct dirent *entry;
44     dirPointer = opendir(directoryPath);
45     if (dirPointer == NULL) {
46         printf("Could not open directory: %s\n", directoryPath);
47         return;
48     }
49     while ((entry = readdir(dirPointer)) != NULL) {
50         char *fileName = entry->d_name;
51         if (strstr(fileName, ".c") != NULL) {
52             printf("%s\n", fileName);
53         }
54     }
55     closedir(dirPointer);
56 }
```

```

58 int main() {
59     int option;
60     char directory[200] = "jobs/"; // Default directory path
61     char job[50];
62
63     do {
64         printf(" Welcome to THE BATCH SIMULATOR\n");
65         printf("(1) List Jobs\n");
66         printf("(2) Set Jobs Directory\n");
67         printf("(3) Compile and run specific program\n");
68         printf("(4) Compile and run all jobs in a specific directory\n");
69         printf("(5) Shutdown\n");
70         printf("(6) List program options\n");
71         printf("(7) Help\n");
72         scanf("%d", &option);
73
74         if (option == 1) {
75             printf("Jobs available in directory: %s\n", directory);
76             listJobs(directory);
77         } else if (option == 2) {
78             printf("Please enter a job directory path:\n");
79             scanf("%s", directory);
80             printf("New job directory set to: %s\n", directory);
81         } else if (option == 3) {
82             printf("Compile and Run a specific program \n");
83             printf("Please enter a program to run: ");
84             scanf("%s", job);
85
86             char compileCommand[200];
87             snprintf(compileCommand, sizeof(compileCommand), "gcc -o \"%s/%s.out\" \"%s/%s.c\"", directory, job, directory, job);
88             system(compileCommand);

```

```

88
89             system(compileCommand);
90
91             char runCommand[200];
92             snprintf(runCommand, sizeof(runCommand), "\"%s/%s.out\"", directory, job);
93             printf("Job output:\n");
94             system(runCommand);
95             printf("Job done!!\n\n");
96         } else if (option == 4) {
97             printf("< Compiling and running all jobs >\n");
98             runAllTasks(directory);
99         } else if (option == 5) {
100             printf("SHUT DOWN!! See Ya!\n");
101             option = 0;
102         } else if (option == 6) {
103             printf("Listing program options ....\n");
104         } else if (option == 7) {
105             printf("----- Help -----");
106             printf("this program allows you to compile and execute C programs from a specified directory. For further assistance, contact Dr. McDowell.\n");
107         } else {
108             printf("Please select a valid option from 1-7.\n");
109         }
110     } while (option != 0);
111
112     return 0;
113 }
```

Now, the Screenshot of the program running successfully is shown below:

On executing the batchmonitor.c, menu gets displayed as below:

```
niraj0010@LAPTOP-VL2JH4U5:/mnt/c/Users/neera/batch$ ./batchmonitor
Welcome to THE BATCH SIMULATOR
1) List Jobs
2) Set Jobs Directory
3) Compile and run specific program
4) Compile and run all jobs in a specific directory
5) Shutdown
6) List program options
7) Help
```

On clicking 1, lists jobs of the default directory which is ./jobs.

```
1
Jobs available in directory: jobs/
dots.c
Factorial.c
hash.c
Summation.c
Welcome to THE BATCH SIMULATOR
1) List Jobs
2) Set Jobs Directory
3) Compile and run specific program
4) Compile and run all jobs in a specific directory
5) Shutdown
6) List program options
7) Help
|
```

On clicking 3, we can compile and run specific program of our choice as shown below:

```
Welcome to THE BATCH SIMULATOR
1) List Jobs
2) Set Jobs Directory
3) Compile and run specific program
4) Compile and run all jobs in a specific directory
5) Shutdown
6) List program options
7) Help
3
Compile and Run a specific program
Please enter a program to run: hash
Job output:
#####
Job done!!
```

On clicking 4, it compiles and runs all the jobs in the directory.

```
Welcome to THE BATCH SIMULATOR
1) List Jobs
2) Set Jobs Directory
3) Compile and run specific program
4) Compile and run all jobs in a specific directory
5) Shutdown
6) List program options
7) Help
```

Now, we had skipped 2. On clicking two, we are given choice to set to different path, we will get outside of our default folder and try to run programs inside the Extra

```
Please enter a job directory path:  
cd /mnt/c/Users/neera/batch/extra  
New job directory set to: cd  
Welcome to THE BATCH SIMULATOR  
1) List Jobs  
2) Set Jobs Directory  
3) Compile and run specific program  
4) Compile and run all jobs in a specific directory  
5) Shutdown  
6) List program options  
7) Help  
Please enter a job directory path:  
New job directory set to: /mnt/c/Users/neera/batch/extra  
Welcome to THE BATCH SIMULATOR  
1) List Jobs  
2) Set Jobs Directory  
3) Compile and run specific program  
4) Compile and run all jobs in a specific directory  
5) Shutdown  
6) List program options  
7) Help  
1  
Jobs available in directory: /mnt/c/Users/neera/batch/extra  
pattern.c
```

```
Welcome to THE BATCH SIMULATOR
1) List Jobs
2) Set Jobs Directory
3) Compile and run specific program
4) Compile and run all jobs in a specific directory
5) Shutdown
6) List program options
7) Help
3
Compile and Run a specific program
Please enter a program to run: pattern
Job output:
Enter the number of rows: 2
*
**
Job done!!
```

No clicking 7, help text with display as:

```
Welcome to THE BATCH SIMULATOR
1) List Jobs
2) Set Jobs Directory
3) Compile and run specific program
4) Compile and run all jobs in a specific directory
5) Shutdown
6) List program options
7) Help
7
----- Help -----
This program allows you to compile and execute C programs from a specified directory. For further assistance, contact Dr. McDowell.
```

On clicking 5, the program will get shut down:

```
Welcome to THE BATCH SIMULATOR
1) List Jobs
2) Set Jobs Directory
3) Compile and run specific program
4) Compile and run all jobs in a specific directory
5) Shutdown
6) List program options
7) Help
5
SHUT DOWN!! See Ya!
niraj0010@LAPTOP-VL2JH4U5:/mnt/c/Users/neera/batch$ |
```