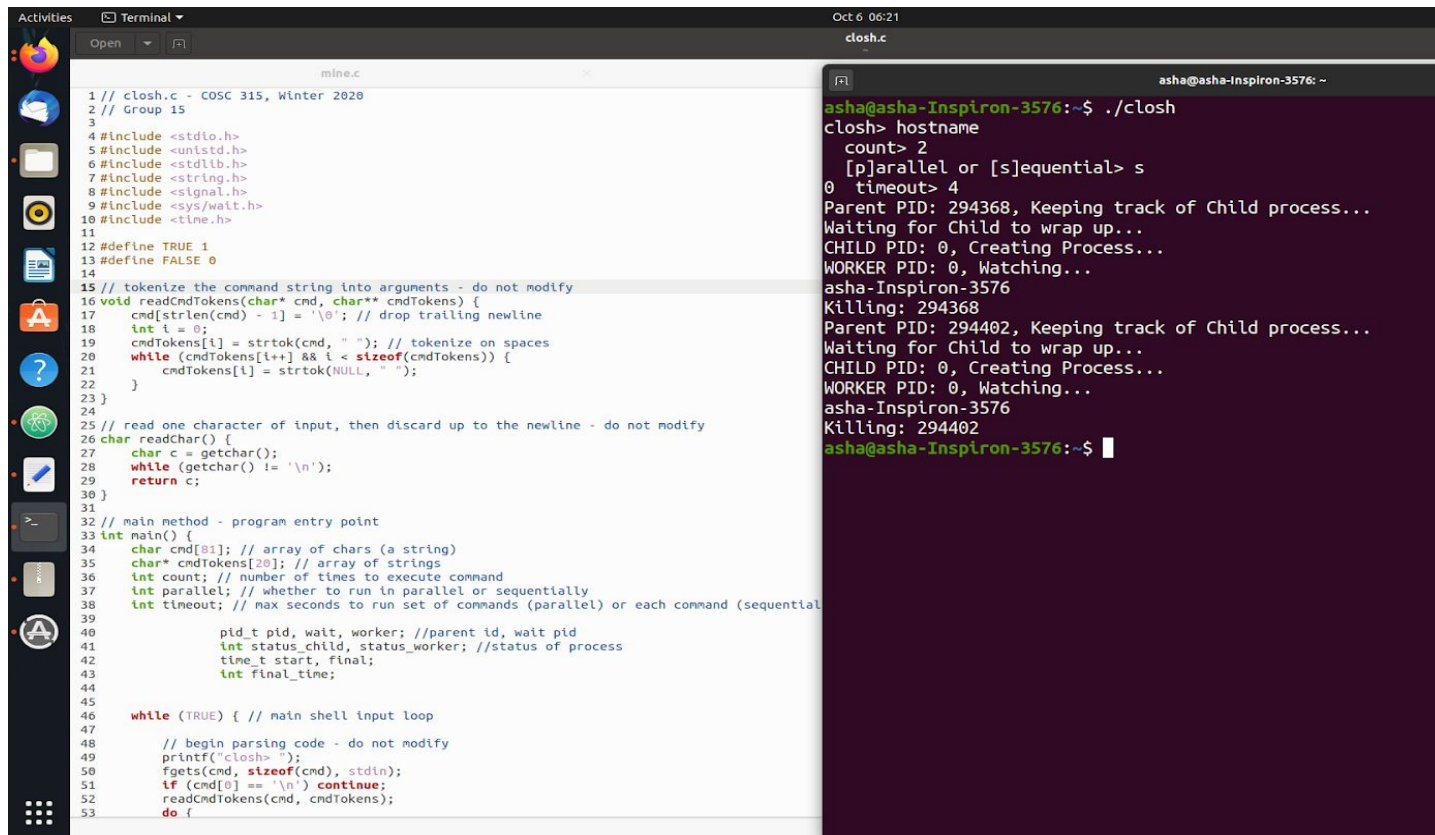


Sample Output From Running Program.



```
1 // cosh.c - COSC 315, Winter 2020
2 // Group 15
3
4 #include <stdio.h>
5 #include <unistd.h>
6 #include <stdlib.h>
7 #include <string.h>
8 #include <signal.h>
9 #include <sys/wait.h>
10 #include <time.h>
11
12 #define TRUE 1
13 #define FALSE 0
14
15 // tokenize the command string into arguments - do not modify
16 void readCmdTokens(char* cmd, char** cmdTokens) {
17     cmd[strlen(cmd) - 1] = '\0'; // drop trailing newline
18     int i = 0;
19     cmdTokens[i] = strtok(cmd, " "); // tokenize on spaces
20     while (cmdTokens[i++] && i < sizeof(cmdTokens)) {
21         cmdTokens[i] = strtok(NULL, " ");
22     }
23 }
24
25 // read one character of input, then discard up to the newline - do not modify
26 char readChar() {
27     char c = getchar();
28     while (getchar() != '\n');
29     return c;
30 }
31
32 // main method - program entry point
33 int main() {
34     char cmd[81]; // array of chars (a string)
35     char* cmdTokens[20]; // array of strings
36     int count; // number of times to execute command
37     int parallel; // whether to run in parallel or sequentially
38     int timeout; // max seconds to run set of commands (parallel) or each command (sequential)
39
40     pid_t pid, wait, worker; //parent id, wait pid
41     int status_child, status_worker; //status of process
42     time_t start, final;
43     int final_time;
44
45     while (TRUE) { // main shell input loop
46
47         // begin parsing code - do not modify
48         printf("cosh> ");
49         fgets(cmd, sizeof(cmd), stdin);
50         if (cmd[0] == '\n') continue;
51         readCmdTokens(cmd, cmdTokens);
52         do {
```

```
asha@asha-Inspiron-3576:~$ ./cosh
cosh> hostname
count> 2
[p]arallel or [s]equential> s
0 timeout> 4
Parent PID: 294368, Keeping track of Child process...
Waiting for Child to wrap up...
CHILD PID: 0, Creating Process...
WORKER PID: 0, Watching...
asha-Inspiron-3576
Killing: 294368
Parent PID: 294402, Keeping track of Child process...
Waiting for Child to wrap up...
CHILD PID: 0, Creating Process...
WORKER PID: 0, Watching...
asha-Inspiron-3576
Killing: 294402
asha@asha-Inspiron-3576:~$
```

ActivitiesTerminalOct 6 06:20

Openmine.c

1 // closh.c - COSC 315, Winter 2020
2 // Group 15
3
4 #include <stdio.h>
5 #include <unistd.h>
6 #include <stdlib.h>
7 #include <string.h>
8 #include <signal.h>
9 #include <sys/wait.h>
10 #include <time.h>
11
12 #define TRUE 1
13 #define FALSE 0
14
15 // tokenize the command string into arguments - do not modify
16 void readCmdTokens(char* cmd, char** cmdTokens) {
17 cmd[strlen(cmd) - 1] = '\0'; // drop trailing newline
18 int i = 0;
19 cmdTokens[i] = strtok(cmd, " "); // tokenize on spaces
20 while (cmdTokens[i++] && i < sizeof(cmdTokens)) {
21 cmdTokens[i] = strtok(NULL, " ");
22 }
23 }
24
25 // read one character of input, then discard up to the newline - do not modify
26 char readChar() {
27 char c = getchar();
28 while (getchar() != '\n');
29 return c;
30 }
31
32 // main method - program entry point
33 int main() {
34 char cmd[81]; // array of chars (a string)
35 char* cmdTokens[20]; // array of strings
36 int count; // number of times to execute command
37 int parallel; // whether to run in parallel or sequentially
38 int timeout; // max seconds to run set of commands (parallel) or each command (sequential)
39
40 pid_t pid, wait, worker; // parent id, wait pid
41 int status_child, status_worker; // status of process
42 time_t start, final;
43 int final_time;
44
45
46
47 while (TRUE) { // main shell input loop
48
49 // begin parsing code - do not modify
50 printf("closh> ");
51 fgets(cmd, sizeof(cmd), stdin);
52 if (cmd[0] == '\n') continue;
53 readCmdTokens(cmd, cmdTokens);
54 do {

closh.c

asha@asha-Inspiron-3576: ~

asha@asha-Inspiron-3576:~\$ gcc closh.c -o closh
asha@asha-Inspiron-3576:~\$./closh
closh> hostname
count> 2
[p]arallel or [s]equential> p
1 timeout> 4
Parent PID: 294165, Keeping track of Child process...
Parent PID: 294166, Keeping track of Child process...
CHILD PID: 0, Creating Process...
CHILD PID: 0, Creating Process...
asha@asha-Inspiron-3576:~\$ ash-Inspiron-3576
asha-Inspiron-3576

```
asha@asha-Inspiron-3576: ~  
asha@asha-Inspiron-3576:~$ ./closh  
closh> hostname  
count> 8  
[p]arallel or [s]equential> s  
0 timeout> 2  
Parent PID: 296742, Keeping track of Child process...  
Waiting for Child to wrap up...  
CHILD PID: 0, Creating Process...  
WORKER PID: 0, Watching...  
Killing: 296742  
Parent PID: 296782, Keeping track of Child process...  
Waiting for Child to wrap up...  
CHILD PID: 0, Creating Process...  
WORKER PID: 0, Watching...  
Killing: 296782  
Parent PID: 296787, Keeping track of Child process...  
Waiting for Child to wrap up...  
CHILD PID: 0, Creating Process...  
WORKER PID: 0, Watching...  
Killing: 296787  
Parent PID: 296793, Keeping track of Child process...  
Waiting for Child to wrap up...  
CHILD PID: 0, Creating Process...  
WORKER PID: 0, Watching...  
Killing: 296793  
Parent PID: 296801, Keeping track of Child process...  
Waiting for Child to wrap up...  
CHILD PID: 0, Creating Process...  
WORKER PID: 0, Watching...  
Killing: 296801  
Parent PID: 296805, Keeping track of Child process...  
Waiting for Child to wrap up...  
CHILD PID: 0, Creating Process...  
WORKER PID: 0, Watching...  
Killing: 296805  
Parent PID: 296811, Keeping track of Child process...  
Waiting for Child to wrap up...  
CHILD PID: 0, Creating Process...  
WORKER PID: 0, Watching...  
Killing: 296811  
Parent PID: 296819, Keeping track of Child process...  
Waiting for Child to wrap up...  
CHILD PID: 0, Creating Process...  
WORKER PID: 0, Watching...  
Killing: 296819  
asha@asha-Inspiron-3576:~$
```