



Sample: C - Assignment in C Language

```
1: #include <stdio.h>
2: #include <stdlib.h>
3: #include <sys/types.h>
4: #include <sys/stat.h>
5: #include <unistd.h>
6: #include <dirent.h>
7: #include <string.h>
8:
9: int main(int argc, char* argv){
10:     // checking arguments
11:     if(argc != 2){
12:         printf("Usage: userproc <u_id>\n");
13:         return 0;
14:     }
15:     // parsing user id from argument line
16:     uid_t uId = atoi(argv[1]);
17:     // validating user id
18:     if(!uId && argv[1][0] != '0'){
19:         printf("Error! Invalid user id\n");
20:         return -1;
21:     }
22:
23:     // opening directory
24:     DIR * dir = opendir("/proc/");
25:     if (!dir){
26:         printf("Error reading /proc directory\n");
27:         return -2;
28:     }
29:
30:     // reading directory entry
31:     struct dirent * entry;
32:     while (entry = readdir(dir)){
33:         // reading only names, that contains numbers (process id)
34:         if(atoi(entry->d_name)){
35:             char * path;
36:             // forming path for each process directory cmdline
37:             asprintf(&path, "/proc/%s/cmdline", entry->d_name);
38:             struct stat buf;
39:             // determining cmdline owner id
40:             if(stat(path, &buf) != 0){
41:                 printf("Error access stat for %s\n", entry->d_name);
42:                 return -3;
43:             }
44:             // owner id must equal our argument id
45:             if(buf.st_uid == uId){
46:                 // if yes, reading program name from cmdline file
47:                 // opens file /proc/<id>/cmdline and reads first word
48:                 FILE* pf = fopen(path, "r");
49:                 if(!pf){
50:                     continue;
51:                 }
52:                 // reading buffer
53:                 char buf[1024];
54:                 if(!fgets(buf, 1023, pf)){
55:                     continue;
56:                 }
57:                 char * tmp;
58:                 asprintf(&tmp, "%s", buf);
59:                 fclose(pf);
60:                 // returning first token
```



Sample: C Programming - Assignment in C language

```
61:         char * cmd = strtok(tmp, " ");
62:         if(cmd){
63:             // if program can access cmdline file, printing process
        id and programm name
64:             printf("%s - %s\n", entry->d_name, cmd);
65:         }
66:     }
67: }
68: }
69: // closing directory
70: closedir(dir);
71: return 0;
72: }
73:
```