Sheet 10

Digdarshan Kunwar

November 2018

Problem 10.1

```c
#include <unistd.h>
int main(int argc, char *argv[])
{
    for (; argc > 1; argc--) {
        if (0 == fork()) {
            (void) fork();
        }
    }
    return 0;
}
```

a)

Case 1 ./foo,
$argc = 1$
• Parent process never loops
• There will be 0 child process as there is no loop

```
Total Process :1  Child Processes:0
```

Case 2 ./foo a,
$argc = 2$
• The main process will be in the loop once
• The child of main will never call the loop again, and the child will call fork() one time in the child process itself
• There will be 2 child processes in total

```
Total Process :3  Child Processes:2
```

Case 3 ./foo a b,
$argc = 3$.
•There will be 2 child process when $argc == 2$, which will have 3 process in total.
•The main will also fork again if it loops.
•In total, $3 * 3 = 9$ process, which means 8 child process .

```
Total Process :9  Child Processes:8
```

We can observe in every iteration the number processes splits into 1 child and another sub child process.And if the child process is not the main parent(the first process) then it is also counted in the child process.So there will be 3 process in total in 1 iteration.
So we can relate it to a formula $TotalProcess = 3^n$ for the total number of processes including the main parent process.
where n is (argc-1).`(argc is argument count)`
Also the number of child processes can be written as $TotalProcess - 1$

Case 4 ./foo a b c,
$argc = 4$
•Here we have total of 3^3 processes.
•Excluding the main parent process we have 26 child process.

```
Total Process :27  Child Processes:26
```

Case 5./foo a b c d,
$argc = 4$
•Here we have total of 3^4 processes.
•Excluding the main parent process we have 80 child process.

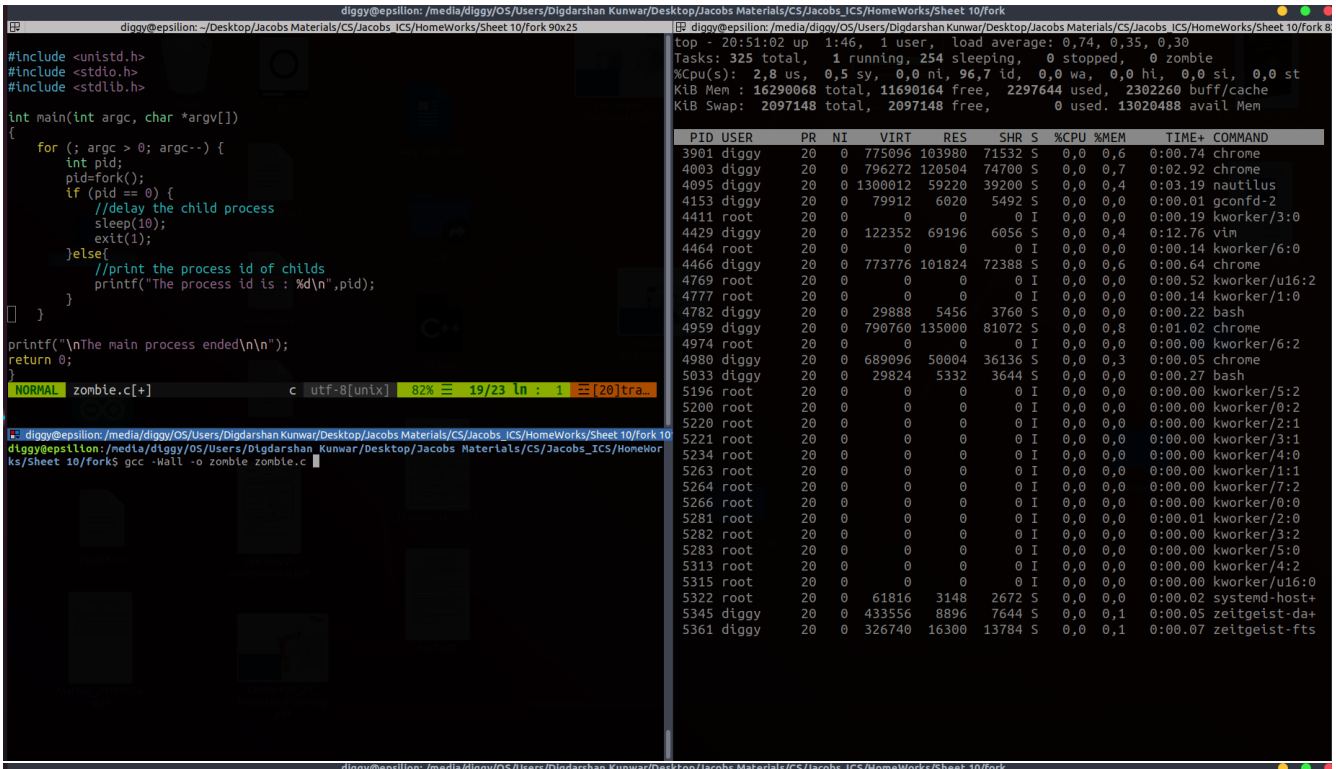
```
Total Process :81  Child Processes:80
```

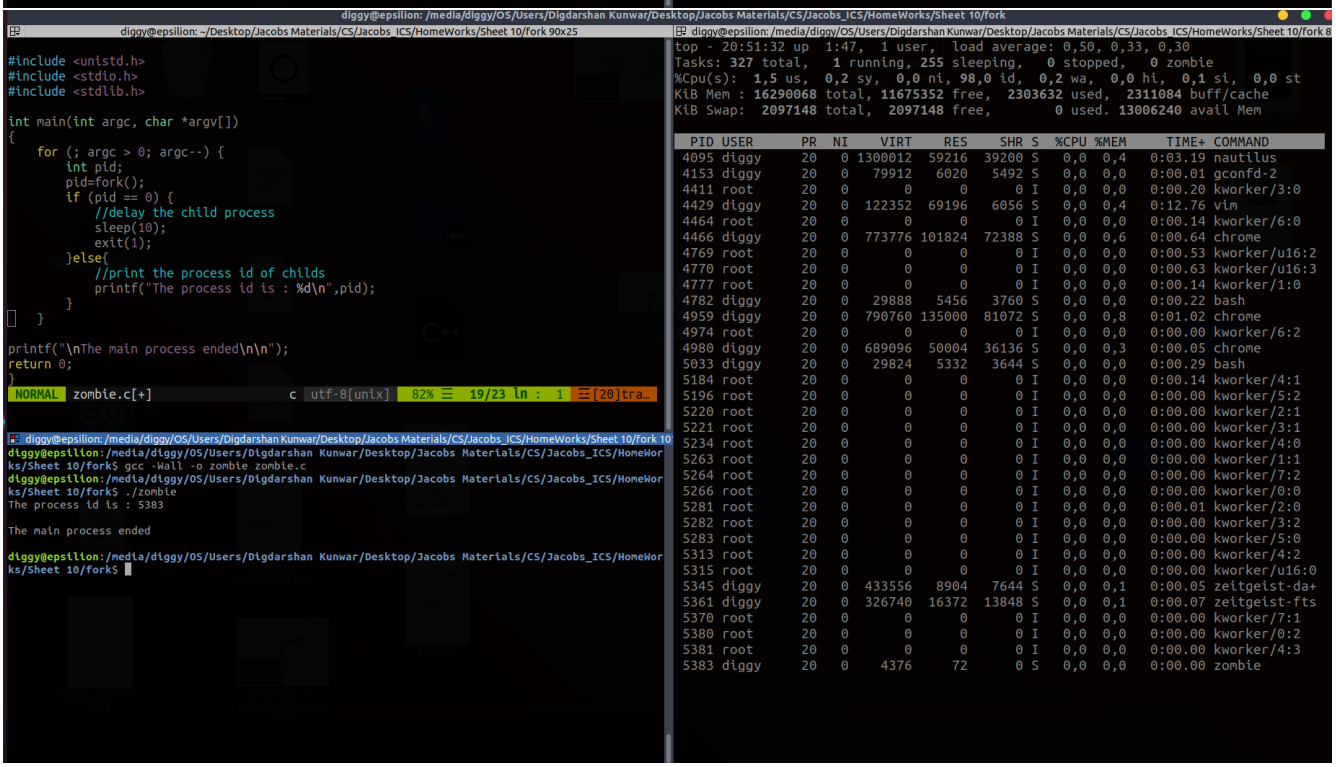
b)

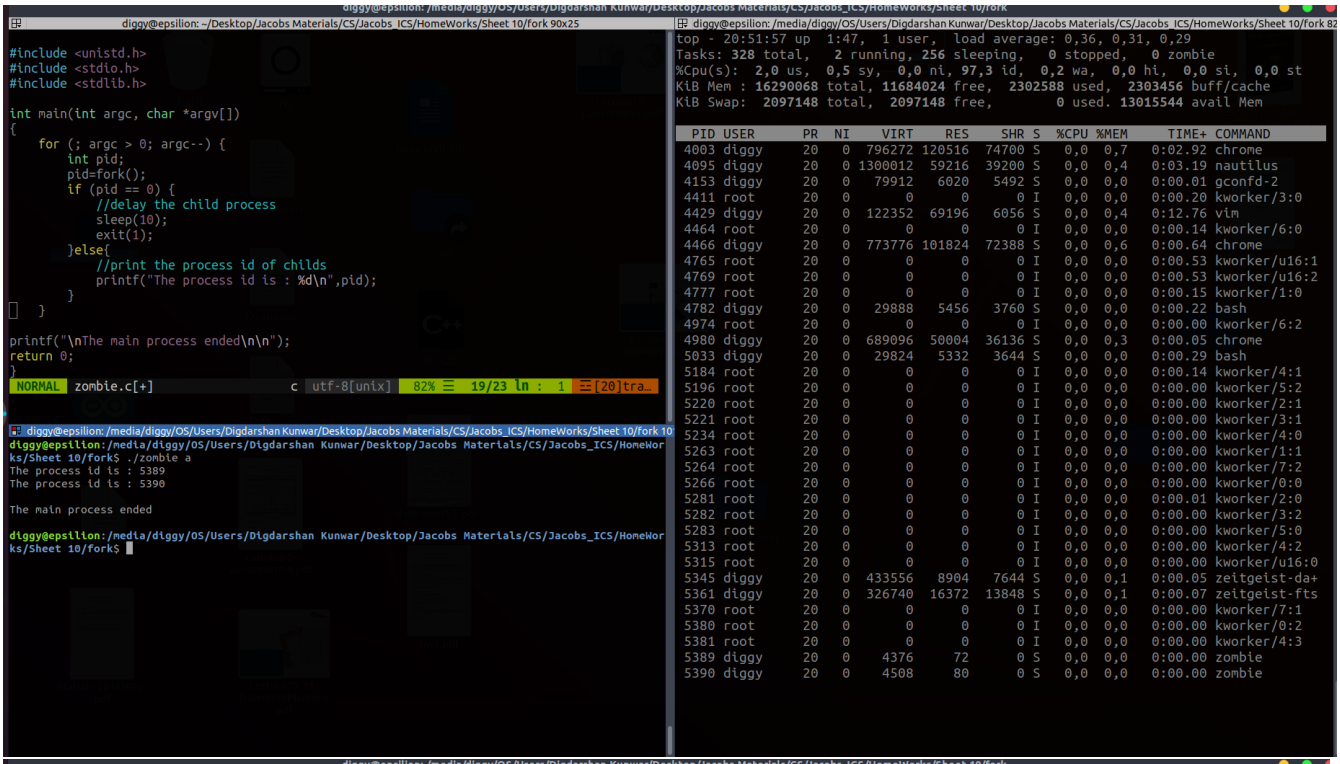
The code is given below :

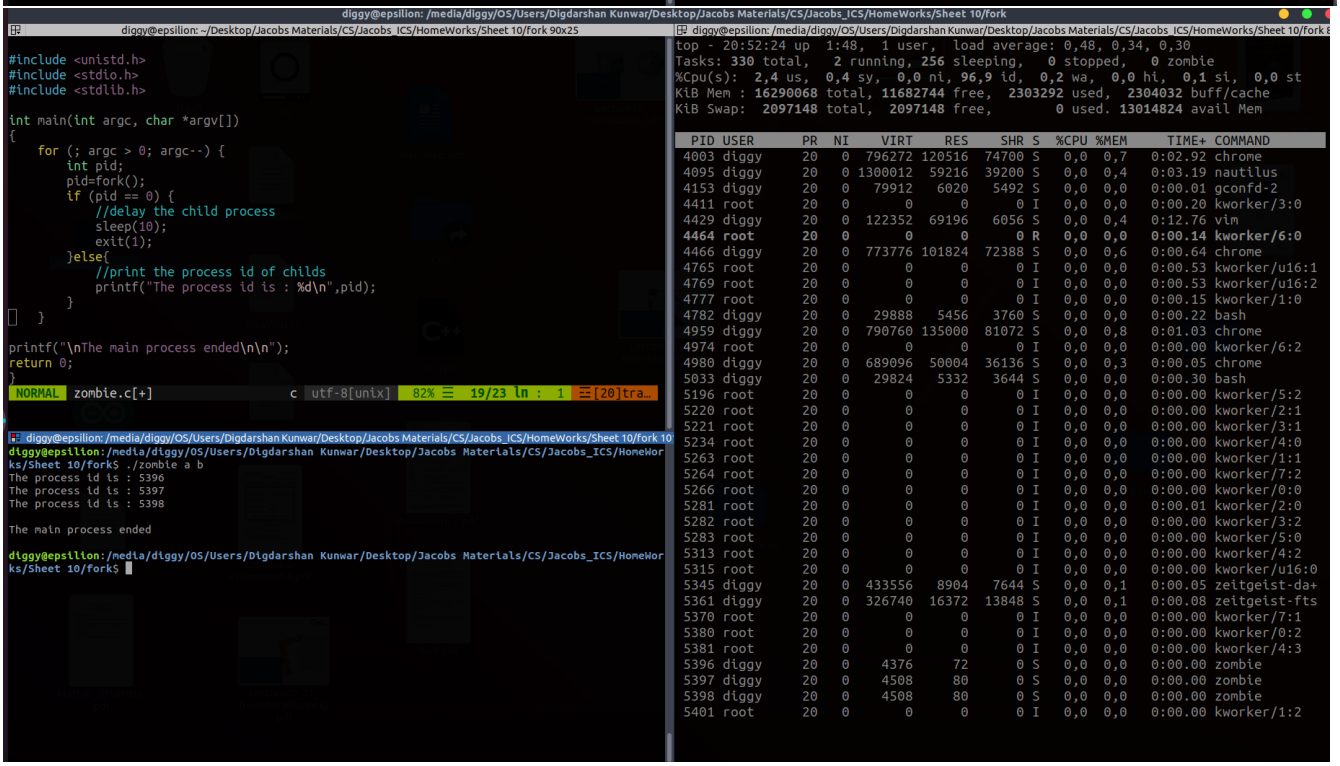
```
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    for (; argc > 0; argc--) {
        int pid;
        pid=fork();
        if (pid == 0) {
          //delay the child process
                    sleep(10);
          exit(1);
        }else{
          //print the process id of childs
          printf("The process id is : %d\n",pid);
        }
    }

printf("\nThe main process ended\n\n");
return 0;
}
```

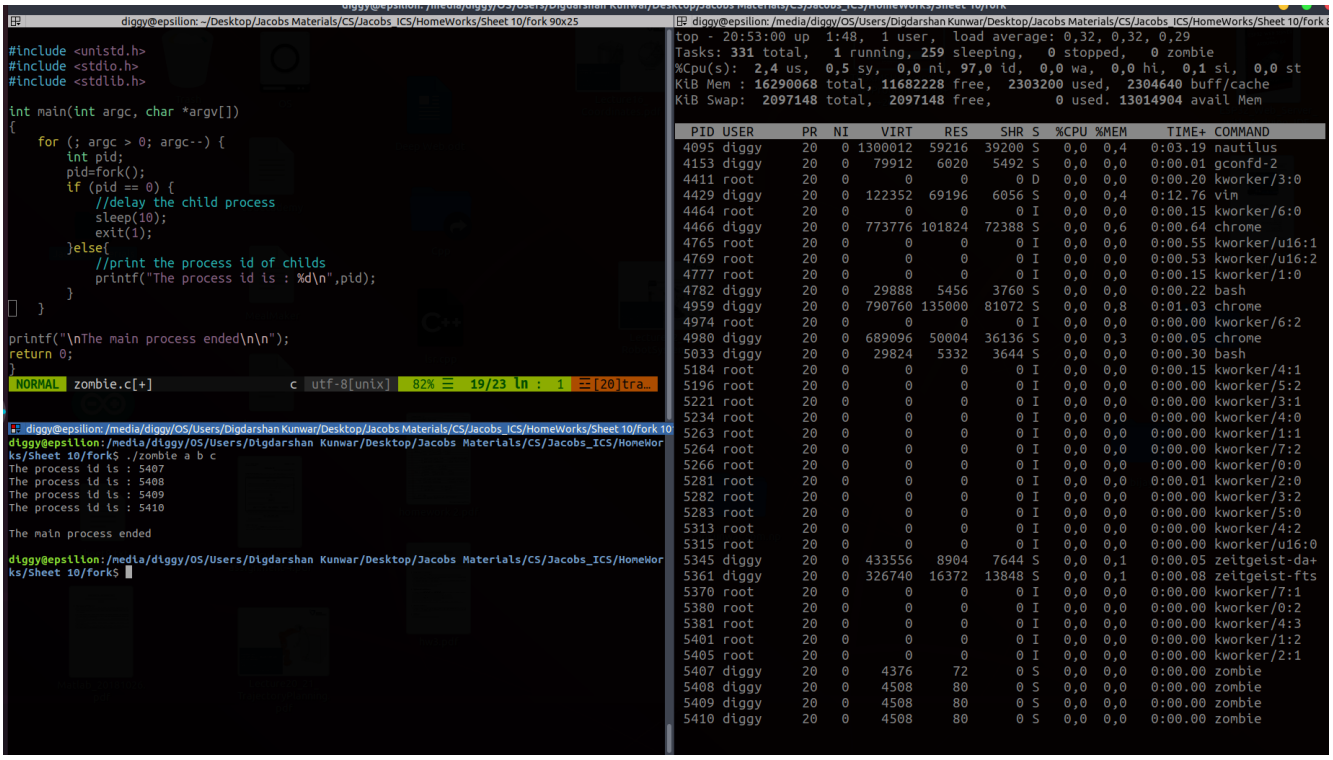
Here the child or the zombie process is still running even if the parent process ends with the message. So we can see the zombie process with the top utility.Because they are still running the sleep() command in the process.
So they can be seen the the process list and we can check for their PID(Process Id).
The implementation is shown in the attached pictures as well.
Creating zombie with different number of arguments :


diggy@epsilion: /media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/Fork
diggy@epsilion: ~/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork 90x25
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
for (; argc > 0; argc--) {
int pid;
pid=fork();
if (pid == 0) {
//delay the child process
sleep(10);
exit(1);
}else{
//print the process id of childs
printf("The process id is : %d\n",pid);
}
}
printf("\nThe main process ended\n\n");
return 0;
}
NORMAL zombie.c[+] c utf-8[unix] 82% 19/23 ln : 1 [20]tra…
diggy@epsilion: /media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork 10
diggy@epsilion:/media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork$ gcc -Wall -o zombie zombie.c
diggy@epsilion: /media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork 8
top - 20:51:02 up 1:46, 1 user, load average: 0,74, 0,35, 0,30
Tasks: 325 total, 1 running, 254 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2,8 us, 0,5 sy, 0,0 ni, 96,7 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 16290068 total, 11690164 free, 2297644 used, 2302260 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used. 13020488 avail Mem
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
3901 diggy 20 0 775096 103980 71532 S 0,0 0,6 0:00.74 chrome
4003 diggy 20 0 796272 120504 74700 S 0,0 0,7 0:02.92 chrome
4095 diggy 20 0 1300012 59220 39200 S 0,0 0,4 0:03.19 nautilus
4153 diggy 20 0 79912 6020 5492 S 0,0 0,0 0:00.01 gconfd-2
4411 root 20 0 0 0 0 I 0,0 0,0 0:00.19 kworker/3:0
4429 diggy 20 0 122352 69196 6056 S 0,0 0,4 0:12.76 vim
4464 root 20 0 0 0 0 I 0,0 0,0 0:00.14 kworker/6:0
4466 diggy 20 0 773776 101824 72388 S 0,0 0,6 0:00.64 chrome
4769 root 20 0 0 0 0 I 0,0 0,0 0:00.52 kworker/u16:2
4777 root 20 0 0 0 0 I 0,0 0,0 0:00.14 kworker/1:0
4782 diggy 20 0 29888 5456 3760 S 0,0 0,0 0:00.22 bash
4959 diggy 20 0 790760 135000 81072 S 0,0 0,8 0:01.02 chrome
4974 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/6:2
4980 diggy 20 0 689096 50004 36136 S 0,0 0,3 0:00.05 chrome
5033 diggy 20 0 29824 5332 3644 S 0,0 0,0 0:00.27 bash
5196 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/5:2
5200 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/0:2
5220 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/2:1
5221 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/3:1
5234 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/4:0
5263 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/1:1
5264 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/7:2
5266 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/0:0
5281 root 20 0 0 0 0 I 0,0 0,0 0:00.01 kworker/2:0
5282 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/3:2
5283 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/5:0
5313 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/4:2
5315 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/u16:0
5322 root 20 0 61816 3148 2672 S 0,0 0,0 0:00.02 systemd-host+
5345 diggy 20 0 433556 8896 7644 S 0,0 0,1 0:00.05 zeitgeist-da+
5361 diggy 20 0 326740 16300 13784 S 0,0 0,1 0:00.07 zeitgeist-fts


diggy@epsilion: /media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/Fork
diggy@epsilion: ~/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork 90x25
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
for (; argc > 0; argc--) {
int pid;
pid=fork();
if (pid == 0) {
//delay the child process
sleep(10);
exit(1);
}else{
//print the process id of childs
printf("The process id is : %d\n",pid);
}
}
printf("\nThe main process ended\n\n");
return 0;
}
NORMAL zombie.c[+] c utf-8[unix] 82% 19/23 ln : 1 [20]tra…
diggy@epsilion: /media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork 10
diggy@epsilion:/media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork$ gcc -Wall -o zombie zombie.c
diggy@epsilion:/media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork$ ./zombie
The process id is : 5383
The main process ended
diggy@epsilion:/media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork$
diggy@epsilion: /media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/Fork
diggy@epsilion: /media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork 8
top - 20:51:32 up 1:47, 1 user, load average: 0,50, 0,33, 0,30
Tasks: 327 total, 1 running, 255 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1,5 us, 0,2 sy, 0,0 ni, 98,0 id, 0,2 wa, 0,0 hi, 0,1 si, 0,0 st
KiB Mem : 16290068 total, 11675352 free, 2303632 used, 2311084 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used. 13006240 avail Mem
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
4095 diggy 20 0 1300012 59216 39200 S 0,0 0,4 0:03.19 nautilus
4153 diggy 20 0 79912 6020 5492 S 0,0 0,0 0:00.01 gconfd-2
4411 root 20 0 0 0 0 I 0,0 0,0 0:00.20 kworker/3:0
4429 diggy 20 0 122352 69196 6056 S 0,0 0,4 0:12.76 vim
4464 root 20 0 0 0 0 I 0,0 0,0 0:00.14 kworker/6:0
4466 diggy 20 0 773776 101824 72388 S 0,0 0,6 0:00.64 chrome
4769 root 20 0 0 0 0 I 0,0 0,0 0:00.53 kworker/u16:2
4770 root 20 0 0 0 0 I 0,0 0,0 0:00.63 kworker/u16:3
4777 root 20 0 0 0 0 I 0,0 0,0 0:00.14 kworker/1:0
4782 diggy 20 0 29888 5456 3760 S 0,0 0,0 0:00.22 bash
4959 diggy 20 0 790760 135000 81072 S 0,0 0,8 0:01.02 chrome
4974 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/6:2
4980 diggy 20 0 689096 50004 36136 S 0,0 0,3 0:00.05 chrome
5033 diggy 20 0 29824 5332 3644 S 0,0 0,0 0:00.29 bash
5184 root 20 0 0 0 0 I 0,0 0,0 0:00.14 kworker/4:1
5196 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/5:2
5220 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/2:1
5221 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/3:1
5234 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/4:0
5263 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/1:1
5264 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/7:2
5266 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/0:0
5281 root 20 0 0 0 0 I 0,0 0,0 0:00.01 kworker/2:0
5282 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/3:2
5283 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/5:0
5313 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/4:2
5315 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/u16:0
5345 diggy 20 0 433556 8904 7644 S 0,0 0,1 0:00.05 zeitgeist-da+
5361 diggy 20 0 326740 16372 13848 S 0,0 0,1 0:00.07 zeitgeist-fts
5370 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/7:1
5380 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/0:2
5381 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/4:3
5383 diggy 20 0 4376 72 0 S 0,0 0,0 0:00.00 zombie


diggy@epsilon: ~/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork 90x25
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
for (; argc > 0; argc--) {
int pid;
pid=fork();
if (pid == 0) {
//delay the child process
sleep(10);
exit(1);
}else{
//print the process id of childs
printf("The process id is : %d\n",pid);
}
}
printf("\nThe main process ended\n\n");
return 0;
}
NORMAL zombie.c[+] c utf-8[unix] 82% 19/23 ln : 1
diggy@epsilon: /media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork
diggy@epsilon:/media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork$ ./zombie a
The process id is : 5389
The process id is : 5390
The main process ended
diggy@epsilon:/media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork$
diggy@epsilon: /media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork
top - 20:51:57 up 1:47, 1 user, load average: 0,36, 0,31, 0,29
Tasks: 328 total, 2 running, 256 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2,0 us, 0,5 sy, 0,0 ni, 97,3 id, 0,2 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 16290068 total, 11684024 free, 2302588 used, 2303456 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used. 13015544 avail Mem
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
4003 diggy 20 0 796272 120516 74700 S 0,0 0,7 0:02.92 chrome
4095 diggy 20 0 1300012 59216 39200 S 0,0 0,4 0:03.19 nautilus
4153 diggy 20 0 79912 6020 5492 S 0,0 0,0 0:00.01 gconfd-2
4411 root 20 0 0 0 0 I 0,0 0,0 0:00.20 kworker/3:0
4429 diggy 20 0 122352 69196 6056 S 0,0 0,4 0:12.76 vim
4464 root 20 0 0 0 0 I 0,0 0,0 0:00.14 kworker/6:0
4466 diggy 20 0 773776 101824 72388 S 0,0 0,6 0:00.64 chrome
4765 root 20 0 0 0 0 I 0,0 0,0 0:00.53 kworker/u16:1
4769 root 20 0 0 0 0 I 0,0 0,0 0:00.53 kworker/u16:2
4777 root 20 0 0 0 0 I 0,0 0,0 0:00.15 kworker/1:0
4782 diggy 20 0 29888 5456 3760 S 0,0 0,0 0:00.22 bash
4974 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/6:2
4980 diggy 20 0 689096 50004 36136 S 0,0 0,3 0:00.05 chrome
5033 diggy 20 0 29824 5332 3644 S 0,0 0,0 0:00.29 bash
5184 root 20 0 0 0 0 I 0,0 0,0 0:00.14 kworker/4:1
5196 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/5:2
5220 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/2:1
5221 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/3:1
5234 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/4:0
5263 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/1:1
5264 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/7:2
5266 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/0:0
5281 root 20 0 0 0 0 I 0,0 0,0 0:00.01 kworker/2:0
5282 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/3:2
5283 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/5:0
5313 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/4:2
5315 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/u16:0
5345 diggy 20 0 433556 8904 7644 S 0,0 0,1 0:00.05 zeitgeist-da+
5361 diggy 20 0 326740 16372 13848 S 0,0 0,1 0:00.07 zeitgeist-fts
5370 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/7:1
5380 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/0:2
5381 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/4:3
5389 diggy 20 0 4376 72 0 S 0,0 0,0 0:00.00 zombie
5390 diggy 20 0 4508 80 0 S 0,0 0,0 0:00.00 zombie


diggy@epsilon: /media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork
diggy@epsilon: ~/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork 90x25
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
for (; argc > 0; argc--) {
int pid;
pid=fork();
if (pid == 0) {
//delay the child process
sleep(10);
exit(1);
}else{
//print the process id of childs
printf("The process id is : %d\n",pid);
}
}
printf("\nThe main process ended\n\n");
return 0;
}
NORMAL zombie.c[+] c utf-8[unix] 82% 19/23 ln : 1
diggy@epsilon: /media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork
diggy@epsilon:/media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork$ ./zombie a b
The process id is : 5396
The process id is : 5397
The process id is : 5398
The main process ended
diggy@epsilon:/media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork$
diggy@epsilon: /media/diggy/OS/Users/Digdarshan Kunwar/Desktop/Jacobs Materials/CS/Jacobs_ICS/HomeWorks/Sheet 10/fork
top - 20:52:24 up 1:48, 1 user, load average: 0,48, 0,34, 0,30
Tasks: 330 total, 2 running, 256 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2,4 us, 0,4 sy, 0,0 ni, 96,9 id, 0,2 wa, 0,0 hi, 0,1 si, 0,0 st
KiB Mem : 16290068 total, 11682744 free, 2303292 used, 2304032 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used. 13014824 avail Mem
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
4003 diggy 20 0 796272 120516 74700 S 0,0 0,7 0:02.92 chrome
4095 diggy 20 0 1300012 59216 39200 S 0,0 0,4 0:03.19 nautilus
4153 diggy 20 0 79912 6020 5492 S 0,0 0,0 0:00.01 gconfd-2
4411 root 20 0 0 0 0 I 0,0 0,0 0:00.20 kworker/3:0
4429 diggy 20 0 122352 69196 6056 S 0,0 0,4 0:12.76 vim
4464 root 20 0 0 0 0 R 0,0 0,0 0:00.14 kworker/6:0
4466 diggy 20 0 773776 101824 72388 S 0,0 0,6 0:00.64 chrome
4765 root 20 0 0 0 0 I 0,0 0,0 0:00.53 kworker/u16:1
4769 root 20 0 0 0 0 I 0,0 0,0 0:00.53 kworker/u16:2
4777 root 20 0 0 0 0 I 0,0 0,0 0:00.15 kworker/1:0
4782 diggy 20 0 29888 5456 3760 S 0,0 0,0 0:00.22 bash
4959 diggy 20 0 790760 135000 81072 S 0,0 0,8 0:01.03 chrome
4974 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/6:2
4980 diggy 20 0 689096 50004 36136 S 0,0 0,3 0:00.05 chrome
5033 diggy 20 0 29824 5332 3644 S 0,0 0,0 0:00.30 bash
5196 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/5:2
5220 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/2:1
5221 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/3:1
5234 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/4:0
5263 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/1:1
5264 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/7:2
5266 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/0:0
5281 root 20 0 0 0 0 I 0,0 0,0 0:00.01 kworker/2:0
5282 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/3:2
5283 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/5:0
5313 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/4:2
5315 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/u16:0
5345 diggy 20 0 433556 8904 7644 S 0,0 0,1 0:00.05 zeitgeist-da+
5361 diggy 20 0 326740 16372 13848 S 0,0 0,1 0:00.08 zeitgeist-fts
5370 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/7:1
5380 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/0:2
5381 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/4:3
5396 diggy 20 0 4376 72 0 S 0,0 0,0 0:00.00 zombie
5397 diggy 20 0 4508 80 0 S 0,0 0,0 0:00.00 zombie
5398 diggy 20 0 4508 80 0 S 0,0 0,0 0:00.00 zombie
5401 root 20 0 0 0 0 I 0,0 0,0 0:00.00 kworker/1:2



Problem 10.2

The code is given below ::

```
#include<iostream>
#include <dirent.h>
#include<string>
using namespace std;

/*
This program recursively shows the content of the current working directory if no
arguments are passed to the main() function of the program. Otherwise, starts a
recursive listing for each of the arguments that are passed to the main() function.
*/

void search (char* name);
int main (int charc , char *charv[]) {
    char* root=new char[1]{'.'};
    if (charc==1){
        search(root);
        delete root;
         return 0;
    }else{
        //if the argc is more than 1
        for (int i =1;i<charc;i++){
            //Calling a search function in main
            search(charv[i]);
        }
    }
```

```
    return 0;
}

//This is the search function
void search (char* name){
    DIR *dir;
    char* fname;
    string loc,dirName;
    struct dirent *current;

    dir=opendir(name);

    //Check if the dir is NULL
    if (dir!=NULL){

        while((current=readdir(dir))!=NULL ){
            //assign fname with the current->d_name
            fname=current->d_name;

            //if the file starts with . then neglect printing it
            if (current->d_name[0] != '.'){

                if (name[0]=='.'&& name[1]=='\0'){
                    cout<<fname<<endl;
                    //location of contents (files) of that directory from current directory
                    loc=fname;
                }else{
                    cout<<name<<"/"<<fname<<endl;
                    if (current->d_type== DT_DIR){
                        //Initial location of the file is name
                        loc=name;
                        //dirName is the filename of a file that is a directory
                        dirName=fname;
                        //loc is location of contents (files) of that directory from
                        current directory
                        loc=loc+"/"+dirName;
                    }
                }
            }
            /*if the file is a directory then we have to read if that directory has
            other files in itself*/
            if (current->d_type== DT_DIR && current->d_name[0] != '.' ){

                //The below steps copy a string to a pointer
                //Copies the loc to char pointer nloc
                char * nloc = new char[loc.size() + 1];
                copy(loc.begin(), loc.end(), nloc);
                nloc[loc.size()] = '\0';

                //Recusively go and read the files in directory
                search(nloc);
                delete nloc;
            }
        }

    }else{
        //If the name doesnt match the directory or if dir is NULL
```

```
    cout<<"Error opening the Directory : "<<name<<endl;
}
//Close the dir
closedir(dir);
}
```