



National Textile University

Department of Computer Science

Subject:

Operating system

Submitted To:

Sir Nasir

Submitted By:

Noor Ul Ain

Registration No:

23-NTU-CS-1221

Lab No:

3 (hometask)

Semester:

Task 1:

Code:

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

int main() {
    pid_t pid;

    pid = fork();

    if (pid < 0) {
        perror("fork failed");
        exit(1);
    }
    else if (pid == 0) {
        printf("Child process (PID: %d) running top...\n", getpid());
        execlp("top", "top", NULL);
        perror("execlp failed");
    }
    else {
        printf("Parent process (PID: %d) created child with PID: %d\n",
getpid(), pid);
        sleep(60);
        printf("Parent process finished.\n");
    }

    return 0;
}
```

Output:

```
File Edit Selection View Go Run ... < > Lab03-hometask [WSL: Ubuntu]
EXPLORER ... C task01.c 1 C task02.c ...
LAB03-HOMETASK [WSL: UBUNTU] Task01 C task01.c Task02 C task02.c ...
PROBLEMS 1 OUTPUT TERMINAL ... Chat + ...
1 #include <stdio.h>
top - 20:28:38 up 4 min, 1 user, load average: 1.04, 0.88, 0.38
Tasks: 49 total, 2 running, 47 sleeping, 0 stopped, 0 zombie
CPU(s): 0.5 us, 0.8 sy, 0.0 ni, 98.3 id, 0.0 wa, 0.0 hi, 0.3 si, 0.0 st
Mem: 3857.2 total, 2220.1 free, 1155.1 used, 621.1 buff/cache
Swap: 1024.0 total, 1024.0 free, 0.0 used. 2702.1 avail Mem
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
705 itsgiv1+ 28 0 11.3g 126388 53128 S 1.3 3.2 0:15.58 node
762 itsgiv1+ 28 0 52.3g 584964 54784 S 1.0 14.8 1:09.93 node
990 itsgiv1+ 28 0 1065888 75332 480600 S 1.0 1.9 0:03.86 node
717 root 28 0 3144 1172 1024 S 0.3 0.8 0:00.08 Relay(718)
718 itsgiv1+ 28 0 1016956 59012 44032 S 0.3 1.5 0:02.32 node
726 root 28 0 3144 1172 1024 S 0.3 0.8 0:00.51 Relay(727)
727 itsgiv1+ 28 0 1018400 59580 44160 S 0.3 1.5 0:01.53 node
1 root 28 0 21644 12252 9180 S 0.0 0.3 0:03.47 systemd
2 root 28 0 3120 2048 1920 S 0.0 0.1 0:00.14 init-systemd(ub
7 root 28 0 3168 2084 1920 S 0.0 0.1 0:04.15 init
61 root 19 -1 42160 15488 14720 S 0.0 0.4 0:00.85 systemd-journal
116 root 28 0 25192 6144 4864 S 0.0 0.2 0:01.16 systemd-udevd
145 systemd+ 28 0 21456 12672 10496 S 0.0 0.3 0:00.44 systemd-resolve
158 systemd+ 28 0 91024 7880 6912 S 0.0 0.2 0:00.24 systemd-timesyn
186 root 28 0 4236 2680 2432 S 0.0 0.1 0:00.08 cron
187 message+ 28 0 9648 4736 4224 S 0.0 0.1 0:00.19 dbus-daemon
202 root 28 0 17964 8192 7424 S 0.0 0.2 0:00.23 systemd-logind
204 root 28 0 1755840 13856 11264 S 0.0 0.3 0:00.34 wsl-pro-service
207 root 28 0 3160 1920 1792 S 0.0 0.0 0:00.09 getty
211 syslog 28 0 22508 5632 4480 S 0.0 0.1 0:00.24 syslogd
213 root 28 0 3116 1792 1664 S 0.0 0.0 0:00.02 getty
226 root 28 0 107000 22400 13184 S 0.0 0.6 0:00.51 unattended-upgr
319 root 28 0 6664 4352 3712 S 0.0 0.1 0:00.04 login
324 itsgiv1+ 28 0 28320 11136 9088 S 0.0 0.3 0:00.66 systemd
327 itsgiv1+ 28 0 21156 3520 1792 S 0.0 0.1 0:00.00 (sd-pam)
366 itsgiv1+ 28 0 6072 4688 3328 S 0.0 0.1 0:00.18 bash
```

Task 2:

```
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>

int main() {
    pid_t pid = fork(); // create child process

    if (pid < 0) {
        perror("fork failed");
        return 1;
    }
    else if (pid == 0) {
        // Child process
        printf("Child (PID: %d) running date command...\n", getpid());
        execvp("date", "date", NULL); // replace process with date
    }
    else {
        // Parent process
        printf("Parent (PID: %d) waiting for child (PID: %d)... \n",
getpid(), pid);
        wait(NULL); // wait for child to finish
    }
}
```

```
        printf("Child finished.\n");
    }

    return 0;
}
```

Output:

The screenshot shows the Microsoft Visual Studio Code interface running in a Windows environment. The code editor displays a C file named `task02.c`. The terminal window shows the execution of the program, where the parent process waits for the child process to finish.

```
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>

int main() {
    pid_t pid = fork(); // Create child process
    if (pid < 0) {
        perror("fork failed");
        return 1;
    }
    if (pid == 0) { // Child process
        printf("Child finished.\n");
        exit(0);
    } else { // Parent process
        int status;
        wait(&status);
        printf("Parent (PID: %d) waiting for child (PID: %d)...", getpid(), pid);
        if (WIFEXITED(status)) {
            printf("Child (PID: %d) exited with status %d\n", pid, WEXITSTATUS(status));
        } else {
            printf("Child (PID: %d) terminated by signal %d\n", pid, WTERMSIG(status));
        }
    }
}
```

Terminal Output:

```
itsgivingtech@DESKTOP-65N72IV:~/Operatingsystemlabs/Lab03-hometask$ ./TASK02
Parent (PID: 1573) waiting for child (PID: 1574)...
Child (PID: 1574) running date command...
Sun Jan 4 20:29:25 PKT 2026
Child finished.
itsgivingtech@DESKTOP-65N72IV:~/Operatingsystemlabs/Lab03-hometask$
```