

OS Lab04

Author: Dipankar Das

Date: 3-2-2022

Roll: 20051554

Question 1

Write a program to create the following process tree

Question 1 (i)

```
  P1
 / | \
P2 P3 P4
```

Solution

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

int main(int argc, char **argv) {
    printf("P1\n");
    for (int i = 1; i < 4; i++)
    {
        pid_t t = fork();
        if (t == 0) {
            printf("P%d\n", i+1);
            exit(0);
        }
    }

    wait(NULL);
    remove(argv[0]);
    return EXIT_SUCCESS;
}
```

Output

```

→ Lab05 git:(master) gcc Q1i.c && ./a.out
P1
P4
P2
P3
Lab05 git:(master) gcc Q1i.c && ./a.out

```

Question 1 (ii)

```

P1
|
P2
|
P3
|
P4

```

Solution

```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <assert.h>
#include <sys/wait.h>

int main(int argc, char **argv) {
    printf("NAME\tPID\tPPID\n");
    for (int i = 0; i < 4; i++) {
        printf("P%d\t%d\t%d\n", i+1, getpid(), getppid());
        pid_t t = fork();
        assert(t >= 0);
        if (t != 0)
            break;
    }

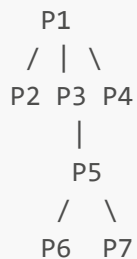
    wait(NULL);
    remove(argv[0]);
    return EXIT_SUCCESS;
}

```

Output

```
→ Lab05 git:(master) cc Q1ii.c && ./a.out
NAME      PID      PPID
P1        3210     1134
P2        3211     3210
P3        3212     3211
P4        3213     3212
```

Question 1 (iii)



Solution

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

int main(int argc, char **argv)
{
    pid_t t = 0;
    printf("NAME\tPID\tPPID\n");
    printf("P1\t%d\t%d\n", getpid(), getppid());
    for (int i = 1; i <= 4; i++)
    {
        t = fork();
        assert(t >= 0);

        if (t == 0) {
            if (i != 1)
                printf("P%d\t%d\t%d\n", i, getpid(), getppid());
            if (i == 3) {
                pid_t tt = fork();
                if (tt == 0) {
                    printf("P5\t%d\t%d\n", getpid(), getppid());
                    for (int i = 6; i < 8; i++) {
                        pid_t tt = fork();

```

```

        if (tt == 0) {
            printf("P%d\t%d\t%d\n", i, getpid(), getppid());
            exit(0);
        }
    }
    wait(NULL);
}
break;
}
}

wait(NULL);
remove(argv[0]);
return EXIT_SUCCESS;
}

```

Output

```

→ Lab05 git:(master) gcc Q1iii.c && ./a.out
NAME      PID      PPID
P1        2711     1134
P3        2714     2711
P4        2715     2711
P5        2716     2714
P2        2713     2711
P7        2719     2716
P6        2718     2716

```

Question 2

Write a C program that will create any given process tree with the following information for each process i.e. level number, index number and the number of children example

```

      P1 {0,0,2}
     /  \
{1,0,0} P2  P3 {1,1,2}
      /  \
{2,0,0} P4  P5 {2,1,1}
          |
        P6 {3,0,0}

```

Solution

```

/**
 * @Assumption
 *   that there are 6 processes
 *   input is given in sorted order i.e. inorder traversal
 * @input
 * 0 0 2
 * 1 0 0
 * 1 1 2
 * 2 0 0
 * 2 1 1
 * 3 0 0
 * @example
 *           P1 {0,0,2}
 *         /  \
 * {1,0,0} P2  P3 {1,1,2}
 *       /  \
 * {2,0,0} P4  P5 {2,1,1}
 *         |
 *         P6 {3,0,0}
 */
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#define NOPROC 6

struct procContent{
    int level, idx, children;
};

struct procContent proc[NOPROC];

void createProcess(int index) {
    int nChild = proc[index].children;
    printf("P%d\t%d\t%d\n", index+1, getpid(), getppid());

    for (int i = 1; i <= nChild; i++) {
        pid_t t = fork();
        if (t == 0) {
            createProcess(index + i);
            break;
        }
    }
    wait(NULL);
}

int main(int argc, char **argv)
{
    printf("Enter the levelNo, indexNO, number of children\n");
    for (int i = 0; i < NOPROC; i++) {
        scanf("%d %d %d", &proc[i].level, &proc[i].idx, &proc[i].children);
    }
}

```

```
}

printf("Process\tPID\tPPID\n");
createProcess(0);

wait(NULL);
remove(argv[0]);
return EXIT_SUCCESS;
}
```

Output

```
→ Lab05 git:(master) gcc Q2.c && ./a.out
Enter the levelno, indexNO, number of children
0 0 2
1 0 0
1 1 2
2 0 0
2 1 1
3 0 0
Process PID      PPID
P1       2737     1134
P2       2738     2737
P3       2739     2737
P5       2741     2739
P4       2740     2739
P6       2742     2741
→ Lab05 git:(master)
```

```

➔ Lab05 git:(master) gcc Q1i.c && ./a.out
P1
P4
P2
P3
➔ Lab05 git:(master) gcc Q1ii.c && ./a.out
P1
P2
P3
P4
➔ Lab05 git:(master) gcc Q1iii.c && ./a.out
NAME      PID      PPID
P1        2711     1134
P3        2714     2711
P4        2715     2711
P5        2716     2714
P2        2713     2711
P7        2719     2716
P6        2718     2716
➔ Lab05 git:(master) gcc Q2.c && ./a.out
Enter the levelNo, indexNO, number of children
0 0 2
1 0 0
1 1 2
2 0 0
2 1 1
3 0 0
Process  PID      PPID
P1      2737     1134
P2      2738     2737
P3      2739     2737
P5      2741     2739
P4      2740     2739
P6      2742     2741
➔ Lab05 git:(master)

```

```
top - 23:18:04 up 25 min, 4 users, load average: 0.31, 0.14, 0.10
Tasks: 159 total, 1 running, 158 sleeping, 0 stopped, 0 zombie
%Cpu0 : 4.0 us, 3.7 sy, 0.0 ni, 92.3 id, 0.0 wa, 0.0 hi, 0.0
%Cpu1 : 1.3 us, 5.4 sy, 0.0 ni, 91.3 id, 0.0 wa, 0.0 hi, 2.0
MiB Mem : 11.7/3898.2 [ 0.3%]
MiB Swap: 0.0/975.0
```

PID	PPID	TIME+	%CPU	%MEM	PR	NI	S	VIRT	RES	UID
2573	2	0:00.69	0.0	0.0	20	0	I	0	0	0
2414	2	0:00.93	0.0	0.0	20	0	I	0	0	0
2413	2	0:00.02	0.0	0.0	20	0	I	0	0	0
2037	2	0:00.03	0.0	0.0	20	0	I	0	0	0
2036	2	0:04.35	2.0	0.0	20	0	I	0	0	0
2030	2	0:02.25	0.3	0.0	20	0	I	0	0	0
1705	1658	1:10.14	5.6	0.1	20	0	S	5808	2364	1000
1658	1657	0:00.17	0.0	0.2	20	0	S	10272	6172	1000
1657	1649	1:46.25	8.6	0.2	20	0	S	15224	6948	1000
1649	561	0:00.17	0.0	0.2	20	0	S	14720	8896	0
1648	1629	0:04.50	0.3	0.1	20	0	R	10112	3664	1000