Lab Assignment-4

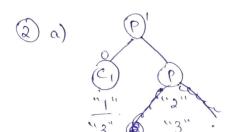
Note: - In all subsequent codes, assume that: Hindude Latdio. 67 # include (stalliboh) # include Zatemistal. h> Hindude Esys/wat. h7 Hincliede (syst types h) header files have been uncluded. (1) a) Run the process where thild process prints "Child" and parent prints "Parent", with their ids. Void main!) pid_t pid = fork(); if (pid == 0) printf("child: "od In", getfick()); while (1); printf ("Parent: oldla", get pides); while (1); b) Terminate the child process. f () reison biov pid_t pid = fork(); if (pid == 0) { print ("Child: Todla", getpid ()); - exit (0);

printf ("Parent: "odla", getprod (1);

else f

while (1);

```
() Torminate the povent process.
   > ( raison bion
       pid-t pid = fork();
       if (pid == 0) {
           printf ("chidd: oldo", getfide);
           while (1);
       else {
           printf ("Parent: olodlo", getfide);
       print (" Pid: olod lo", getpid ());
 d) Make fount process wait for child to complete its task.
  f ( ) risner biou
      pid - t pid = fork();
       if (pid == 0) { printf ("chidd: olodla", get pide)); }
       else &
            print ("Parent: Todlo", get pid ());
            wait(NULL);
        printf ("Pid: of od lo", get pid());
   e) Terminate child process.
   7 () riam biou
        pid-t pid=fockes;
        if (pid == 0) {
             printf ("Child: "lod in", getpid()),
             - exit(0);
             printf ("Parent: % dlm", getpides);
             wait (NULL);
        printf ("Pid: %d/m" = get pide);
```



0[1:2313.

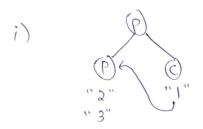
- O[P=123
- c) i=5 (P)
- ofp = Child: 6 Parent: 6

- ofp: Child: 6 Parent: 7
- e) i=5
 (p)
 (c)
 i=5
 i=5+1=6
- of p: Child: 6 Parent: \$5
- f) i= S

 i= 6

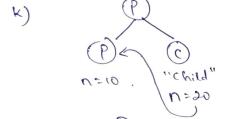
 i= S+1= 6
- off: Child: 6 Parent: 6
- g) i=5 P
- ofp: child: 5 Parent: 5

Off: Child: 5 Parent: 5

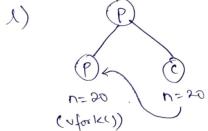


0 (1:123

0/19: 123

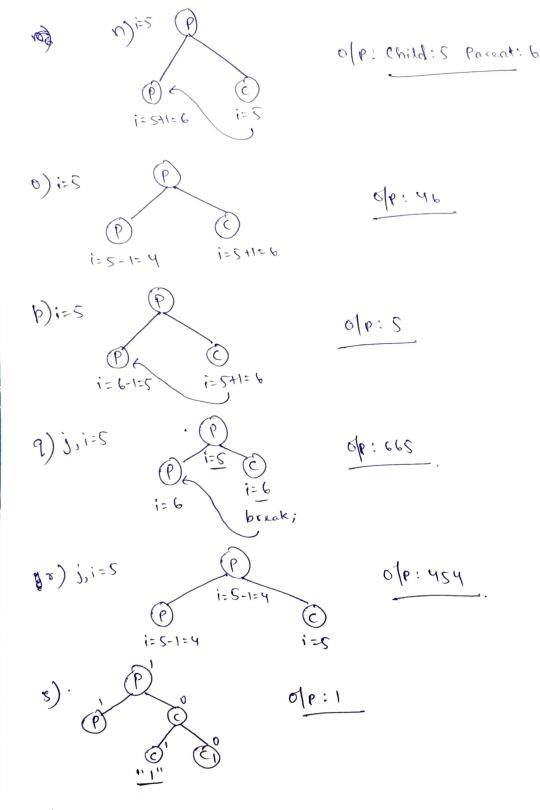


ofe: child n=20 Parent n=10



olp: Child n=20 Porent n=20

0/6: 6 6 6 6



& t) Next bage

t) funl()

". 4 processes will be returned ito main().

main ()

```
(3) was to create 3 child processes to:
          - copy contents of file to filed
          - display contents of filed
         - display souled content of file 2 in revene order.
() enain ()
   if (fork() 20) {
          Sleep(1);
    else §
        printf (" child I id: old Pasent id: old In", get pide), get ppide));
        printf ("Into Copy the contents of fletxt to factatio");
        exectp ("cp", "cp", "-f", "fi.txt", "fa.txt", NULL),
    if (fork()>0)
         · Sleep (1);
    else ?
        printf ("Child 2 id: Tod Parent id: Tod In", getpide), getppide))
        printf ("InInDisplay the contents of faction");
         execlp ("cat", "cat", "fa.txt", NULL);
     if (fock() > 0)
          sleep(1);
     else f
        printf ("Child 3 id: %d Porent id: of.dln", getpides, getphides);
         pernif ("InInSort the contents of 12. Extin")
         exectly ("sort", "sort", "-", "fa.txt", NULL);
    retwo 0;
```

```
(4) Wat to generate a Liberacci series and stone it in an avery,
    and then display prime numbers in the series.
int main () &
    int sz;
    printf (" Enter the length of the fibonacci series:");
     scanf (" dod", &sz);
     int aver [sz];
     of (ofork() == 0) }
         int m1=0, m2=1;
         for (int i=0; i < 82; i++)
            :100 = [i] row
            W3=201+203;
            w1 = wg1 - m1?
          - exit(0);
      else &
          wait (NULL);
          printf ("downey of "led length of fibonacci series is:", sz);
          for (int i=0; i < 82; i+1)
                frint ("olod", au [i]),
          print("1 ");
          int macount;
          printf ("Prime numbers at different locations: 10");
          7(++i; 28 >i; 1=i tri) rop
              Count = 1;
               for (m=2; m< wor [i]; m++) }
                   if (avr[i] % m == 0) count ++;
               if (count ==1)
                   frints ("dod is a prime fibonace number at Modin", avrili]; iti);
         setwon 0;
```