## P.S. Assignment -1

- Respond the Aliver a company the events a shild process using forthe system start the area of the burd at the species of the burd and the second the foot stimple and amindress must be regulated and there are a supplicable that there is a second the burd that "there's aga even at the third aliver and them.
  - a) Run the foregoian and trace the state of hoth forecess.
  - b) Lemnate the child process then trace the state of forocessors.
  - c) Run the foregram and brace the state of both forecesses, terminate the parent forecess. Then trace the state of fracesses.
  - d) hodify the freguen so that the parent frecess after displaying the nessage will wait for child forcess to complete its task. Again own the program and trace the state of both forcesses.
  - e) turninate the child process. Then trace the state of processes.

```
Priogram:

# include < setio. n>

# include < sep/ lights. n>

# include < sep/ wail. n>

int nam() {

l (forte() == 0) {

fruitf (" child %d \n", getfrid());

unite (1);

}

else {

fourtly (" Parent %d \n", get frid());

wait (NULL);

unite (1): }
```

```
return o;
Void ac) {
   if (forhc) = =0) {
       fourth ("child: "od m", getfrid ());
    else
       fruit ("Parent: &d \n", getfrid ());
3
void bc) 3
    4 (force c) ==0) &
        frunty ("child: "d \n", getfrid ());
         -exit (0);
   3
       found ("Pavent: "od m", get fied ());
}
void cc) {
   if (frech ()==0) }
         fountly ("child: "od \n", getfid());
    else 3
       frunty ("Povent: %d \n", getfid());
       - exect (0);
void de) {
    if (forle () == 0) &
         founty ("shild: %d m", get fied ());
```

else ?

```
hound ("Parend: od \n", getfrid ());
      wait (NULL);
  3
3
void ec) {
     26 (forter) ==0) }
          fruntly ("child: "d \n", get fid());
           - esul (0);
      else 3
         founts ("Pount: %d \n", getfid ());
          wait (MULL);
      3
int man () &
    ac);
    b();
     c();
     der;
     cc);
     setem 0;
 }
output :-
 o) Parent: 29171
     child: 29172
 6) Parent: 29 8.18
```

child: 29219

```
c) Parent: 29491
child: 29492
```

( & ) Deace the output of the following program segment:

```
a) und man()

2 (fortic) = = 0)

fountly ("1");

else

fountly ("3");

fruntly ("3");

return 0;

}
```

output :- 2313

```
b) int main ()

if (if order() = = 0)

fountly ("1");

else
fountly ("3");

fountly ("3");
```

output: - 1323

timeout: the monitored command dumped core.

```
(minon bin (a
    hid-t fid;
     ml l = 5;
     find = fook();
     e= i+1;
      if (frid ==0)
         fourth ("child: "d", i);
     else
         wand (NULL);
        founty ("Povend: %d; 1);
      return 0;
  3
              shild: 6
 outful :-
              Parent: 6
d) int main ()
   E fid-t fid;
     ind 1=6;
     frid = V forche ();
      1=1+1;
     of (fid = = 0)
     2 fountly ("child: %d", 1);
       -esul(0);
      else
        founty ("Parent: %d", ();
      returno;
```

```
6
```

```
outful :-
              child: 6
              Povent: 7
 e) und mann ()
  { hid-t hid;
     int &= 5;
      fid = foele();
      if (fid ==0)
      2 0 = e+1;
         frenty (" child: "6d", i);
      3
      else
      E wait (NVLL);
        fount (" Powerd: od"; i);
     seturn o;
            child: 6
 output: -
              Parent: 5
f) und main()
     fid-t fid;
     int 1 = 5;
     frid = vforke);
      if (fid = = 0)
       e=i+1;
       fount (" child: "od", i);
        -esut (0);
```

```
else &
      fruith ("Parent: "d', i);
     returno;
 3
            chied: 6
            Parent: 6
9) int main ()
  2 ml l=5;
     if (fork () == 0)
       foundy (" whild: "d", 1);
      else
      freinth ("Parent: "d", ");
       return 0;
   outful: - Parent: 5
               wild: 5
h) int main ()
   2 in 1 = 5;
     ef (vforte () = = 0)
                                          outful : - child: 5
     2 fruith (" child: "/od", 1);
                                                     Parent: 5
         ·enút (0);
       else
        frently (" Pount: "/od", i);
      ; Omenter
```

```
i) int main ()
     2 " (forth() = = 0)
         dee
             ward (NULL);
             frants (" 2");
             ("E") floring
          outwern 0,
    ?
i) int maine)
    قل ( ( المحمود ) = = 6 )
          fruith ("1");
           -exul(0) ;
        else
Z
           fourth ("2");
          (0 moutere
(x) and mamc)

find t e1;
```

int n = 10 ;

c1 = foule();

```
in (c1 = = 0)
      fruity (" will ("");
       n = 20
       founty (" n = "od \n", n);
     3
                                            outpal: -
      clse
      2
        wait (NULL);
                                                        Parent
                                                        n=10
         frenth ("Parend \n");
         forint (" n = %d \n ",n);
        return 0:
1) int main (1
  fid-t cl+
    int n = 10;
    a = vforke);
    4 (d ==0)
                                          outful : child
                                                       n = 20
       frunt ("child m");
                                                      Parent
        n=20;
                                                       n = 20
        freuntly (" n = %d \n",n);
         - exit (6);
       else
          frunt ("Parent \n");
          frund (" n = %d \n", n);
       return 0;
```

3

```
10
```

```
m) int main ()
    2 ind 1=5;
      forth ();
                                                 6666
       i= 2+1;
       force ();
       Hours (stdere, " ", d", e);
      return 0;
 n) int main()
    E fid-t fid;
      int & = 5;
      frid = yforde ();
       If (frid = = 0)
         frunts ("-child: 1/0 d", ");
         - excit (0);
                                        output: - whild: 5
       3
                                                   Parent: 9
        else
          0 = 0+1;
          fountly ("Parent: "d", ");
       returno;
0) int main()
  2 mt 1=5;
    if (forth () = = 6)
```

```
11
```

```
else
        1=1-1;
                                          outful: - 46
        frountly (stateur, "%d", ");
        : o novelese
  }
 () int main()
       int 6=5;
       2f (vforthe) = =0)
       ٤
٤= ١+1;
       - excet (0);
        else
          1 = 1+1;
       forunty (stolere, ""/od", i);
       seturn 0;
    z
q) int main ()
   2 mt {,i=5;
      for(g-1; g<3; g+t)
      2 if ( forthe) = =0)
        ?
%= %+1;
           break;
        else
           wait (KIVLL);
      I ffount (stoler, "%a", i); return o;
```

```
e) int mains
   2 wit j,i=5
      fore ( f= 1; j< 3; j+t)
        if (facec) != 0)
                                          authul ? - 4 4 5
         2 0= 1+1;
            weak;
       3
       frounty estolerer, "%d", i);
       ; o mentere
    3
8) mt mamc)
   2 if ( foods () = =0)
                                        outful :- 1
       ef (forte())
          found (" 1 \n");
      : 0 mentere.
t) void fund () {
       forte();
                                      outfuil :-
        forter);
        fountly ("1 \n");
    Ent main () &
        fund ();
        fount ("1 \n");
```

return 0;

a3) Trace the following freguram segment & determine how many forocesses are vicated. Decaw a graph that shows how the processes are related.

a) int main() Ž. 26 (fork () 82 forte ()); fourth ("1"); setum 0;

b) int main () if (fork () 11 fork ()); fourth ("1"); setum 0;

output :- 1 1 1 Pount (111) => 0/P = [11]

outful: -111 Pount (211) => 0/P= [111

() man bre (a fid-4 c1, C2; C2 =0; C1 = fork(); if (c1 = =0) C2 = forle(); if (e2>0) forle (); frunty ("1"); return 0;

output ?- 1111 C2 = 0 found (1) found (2) (1) brively

```
output:
                                                                           14
                                                1111
d) int main ()
    fud-t c1=1, c2=1;
    cs = forte();
                                        C2=1
                                                    © C,=0
    A (c7 / = 0)
        c2 = fortel);
     if (c2 = = 0)
         forter);
                                                                   fruit (1)
                                    print(4)
                                              fried (4)
                                                        found (1)
     fourth ("1");
                                                  0/0= 1111
     suburn 0;
                                        outful :- 11111
e) wit main )
     of (forker) 11 forker)
           forte ();
      fruit ("1");
      seturn 0;
                                                                 fruid (1)
                                            (1)
                                      (1)
                                             0/0= 11111
                                                            9 2 2 3
                                                        2
                                                   2 3
f) int main ()
                                      outful : -
    il Cforh() 88 (1, forh()))
        if (forh() 11 yould))
            forte ();
                                                                     freinel (3)
                                                      frint (1)
                                           (1) House
      founty ("2");
                                                                  fourth (2)
                                                         fairl(2)
     return 0;
                                                             2 2 2
                                                     222
                                                    Scanned with CamScanner
```