

main.c — ad_pr2020

C main.c

act_colab_03 > C main.c > main(int, char * [])

```
10  /*-----*/
11  /* [START] C Libraries */
12  #include <stdio.h>
13  #include <stdlib.h>
14  #include <unistd.h>
15  #include <sys/types.h>
16  #include <sys/stat.h>
17  #include <fcntl.h>
18  #include <string.h>
19  #include <dirent.h>
20  #include <time.h>
21  #include <pwd.h>
22  #include <grp.h>
23  #include <limits.h>
24  int main(int argc, char* argv[])
25  {
26      return 0;
27  }
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

1: zsh

m4uz@MacBook-Pro-de-Irving ad_pr2020 %

```

C main.c x
act_colab_03 > C main.c > main(int, char * [])
12  #include <stdio.h>
13  #include <stdlib.h>
14  #include <unistd.h>
15  #include <sys/types.h>
16  #include <sys/stat.h>
17  #include <fcntl.h>
18  #include <string.h>
19  #include <dirent.h>
20  #include <time.h>
21  #include <pwd.h>
22  #include <grp.h>
23  #include <limits.h>
24  int main(int argc, char* argv[])
25  {
26      int pid;
27      if (argc != 1) {
28          fprintf(stderr, "usage: %s\n", argv[0]);
29          return -1;
30      }
31      if ( (pid = fork()) < 0 ) {
32          perror("fork");
33          return -2;
34      } else if (pid == 0) {
35          printf("id = %i - CHILD PROCESS\n", getpid());
36          sleep(2);
37      }
38  }

```

```

m4uz@MacBook-Pro-de-Irving ad_pr2020 %

```

main.c — ad_pr2020

C main.c

act_colab_03 > C main.c > main(int, char * [])

```
12 #include <stdio.h>
13 #include <stdlib.h>
14 #include <unistd.h>
15 #include <sys/types.h>
16 #include <sys/stat.h>
17 #include <fcntl.h>
18 #include <string.h>
19 #include <dirent.h>
20 #include <time.h>
21 #include <pwd.h>
22 #include <grp.h>
23 #include <limits.h>
24 int main(int argc, char* argv[]){
25     int pid, num;
26     if (argc != 2) {
27         fprintf(stderr, "usage: %s number\n", argv[0]);
28         return -2;
29     }
30     num = atoi(argv[1]);
31     if (num <= 1) {
32         fprintf(stderr, "%s: the parameter must be a positive integer number\n", argv[0]);
33         return -2;
34     }
35     // printf("%d", num);
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

1: zsh

```
m4uz@MacBook-Pro-de-Irving act_colab_03 % gcc main.c -o main
m4uz@MacBook-Pro-de-Irving act_colab_03 % ./main
m4uz@MacBook-Pro-de-Irving act_colab_03 % gcc main.c -o main
m4uz@MacBook-Pro-de-Irving act_colab_03 % ./main
usage: ./main number
m4uz@MacBook-Pro-de-Irving act_colab_03 %
```

C main.c

act_colab_03 > C main.c > main(int, char * [])

```

15 #include <sys/types.h>
16 #include <sys/stat.h>
17 #include <fcntl.h>
18 #include <string.h>
19 #include <dirent.h>
20 #include <time.h>
21 #include <pwd.h>
22 #include <grp.h>
23 #include <limits.h>
24 void child_process() {
25     srand( getpid() );
26     printf("PID = %i, PPID = %i, SLEEP = %i\n", getpid(), getppid());
27     sleep(1);
28     printf("PID = %i, PPID = %i\n", getpid(), getppid());
29     exit(0);
30 }
31 int main(int argc, char* argv[]){
32     int pid, num;
33     if (argc != 2) {
34         fprintf(stderr, "usage: %s number\n", argv[0]);
35         return -2;
36     }
37     num = atoi(argv[1]);
38     if (num <= 1) {

```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

1: zsh

```

m4uz@MacBook-Pro-de-Irving act_colab_03 % gcc main.c -o main
m4uz@MacBook-Pro-de-Irving act_colab_03 % ./main 12.14
m4uz@MacBook-Pro-de-Irving act_colab_03 % ./main -10
./main: the parameter must be a positive integer number
m4uz@MacBook-Pro-de-Irving act_colab_03 %

```

C main.c

act_colab_03 > C main.c > child_process(int *, int *)

```
20 #include <time.h>
21 #include <pwd.h>
22 #include <grp.h>
23 #include <limits.h>
24 void child_process(int *i, int *n_actual) {
25     int pid;
26     if (n_actual == i)
27     {
28         srand( getpid() );
29         printf("PID = %i, PPID = %i NIVEL = %i\n", getpid(), getppid());
30         sleep(1);
31         printf("PID = %i, PPID = %i\n", getpid(), getppid());
32         exit(0);
33     }
34 }
35 int main(int argc, char* argv[]){
36     int pid, num;
37     if (argc != 2) {
38         fprintf(stderr, "usage: %s number\n", argv[0]);
39         return -2;
40     }
41     num = atoi(argv[1]);
42     if (num <= 1) {
43         fprintf(stderr, "%s: the parameter must be a positive integer number\n", argv[0]);
44         return -2;
45     }
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

```
m4uz@MacBook-Pro-de-Irving act_colab_03 % gcc main.c -o main
m4uz@MacBook-Pro-de-Irving act_colab_03 % ./main 12.14
m4uz@MacBook-Pro-de-Irving act_colab_03 % ./main -10
./main: the parameter must be a positive integer number
m4uz@MacBook-Pro-de-Irving act_colab_03 %
```

1: zsh

C main.c

act_colab_03 > C main.c > child_process(int *, int *)

```
20 #include <time.h>
21 #include <pwd.h>
22 #include <grp.h>
23 #include <limits.h>
24 void child_process(int *i, int *n_actual) {
25     int pid;
26     if (n_actual == i)
27     {
28         srand( getpid() );
29         printf("PID = %i, PPID = %i NIVEL = %i\n", getpid(), getppid());
30         sleep(1);
31         printf("PID = %i, PPID = %i\n", getpid(), getppid());
32         exit(0);
33     }
34 }
35 int main(int argc, char* argv[]){
36     int pid, num;
37     if (argc != 2) {
38         fprintf(stderr, "usage: %s number\n", argv[0]);
39         return -2;
40     }
41     num = atoi(argv[1]);
42     if (num <= 1) {
43         fprintf(stderr, "%s: the parameter must be a positive integer number\n", argv[0]);
44         return -2;
45     }
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

```
m4uz@MacBook-Pro-de-Irving act_colab_03 % gcc main.c -o main
m4uz@MacBook-Pro-de-Irving act_colab_03 % ./main 12.14
m4uz@MacBook-Pro-de-Irving act_colab_03 % ./main -10
./main: the parameter must be a positive integer number
m4uz@MacBook-Pro-de-Irving act_colab_03 %
```

1: zsh

C main.c

act_colab_03 > C main.c > child_process(int *, int *)

```

12 #include <stdio.h>
13 #include <stdlib.h>
14 #include <unistd.h>
15 #include <sys/types.h>
16 #include <sys/stat.h>
17 #include <fcntl.h>
18 #include <string.h>
19 #include <dirent.h>
20 #include <time.h>
21 #include <pwd.h>
22 #include <grp.h>
23 #include <limits.h>
24 void child_process(int *i, int *n_actual) {
25     int pid;
26     if (n_actual == i)
27     {
28         srand( getpid() );
29         printf("PID = %i, PPID = %i NIVEL = %i\n", getpid(), getppid());
30         sleep(1);
31         printf("PID = %i, PPID = %i\n", getpid(), getppid());
32         exit(0);
33     }
34     else
35     {

```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

1: zsh

```

m4uz@MacBook-Pro-de-Irving act_colab_03 % gcc main.c -o main
m4uz@MacBook-Pro-de-Irving act_colab_03 % ./main 12.14
m4uz@MacBook-Pro-de-Irving act_colab_03 % ./main -10
./main: the parameter must be a positive integer number
m4uz@MacBook-Pro-de-Irving act_colab_03 %

```

```
act_colab_03 > C main.c >  child_process(int *, int *)
```

1: zsh

```
m4uz@MacBook-Pro-de-Irving act_colab_03 % gcc main.c -o main
m4uz@MacBook-Pro-de-Irving act_colab_03 % ./main 12.14
m4uz@MacBook-Pro-de-Irving act_colab_03 % ./main -10
./main: the parameter must be a positive integer number
m4uz@MacBook-Pro-de-Irving act_colab_03 %
```


C main.c

act_colab_03 > C main.c > main(int, char * [])

```
51     wait(NULL);
52 }
53 }
54     exit(0);
55 }
56 }
57 int main(int argc, char* argv[])
58 {
59     int pid, num;
60     if (argc != 2) {
61         fprintf(stderr, "usage: %s number\n", argv[0]);
62         return -2;
63     }
64     num = atoi(argv[1]);
65     if (num <= 1) {
66         fprintf(stderr, "%s: the parameter must be a positive integer number\n", argv[0]);
67         return -3;
68     }
69     child_process(&num, 0);
70     return 0;
71 }
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

```
m4uz@MacBook-Pro-de-Irving act_colab_03 % gcc main.c -o main
m4uz@MacBook-Pro-de-Irving act_colab_03 % ./main 12.14
m4uz@MacBook-Pro-de-Irving act_colab_03 % ./main -10
./main: the parameter must be a positive integer number
m4uz@MacBook-Pro-de-Irving act_colab_03 %
```

1: zsh

C main.c

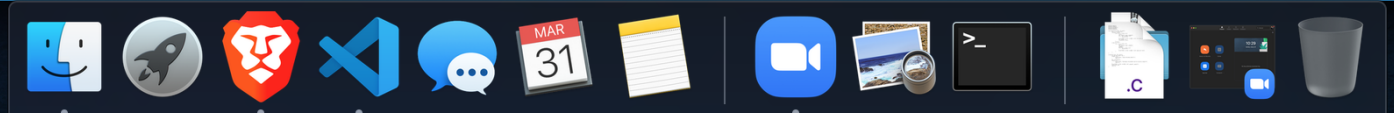
```
act_colab_03 > C main.c > child_process(int, int)
12 #include <stdio.h>
13 #include <stdlib.h>
14 #include <unistd.h>
15 #include <sys/types.h>
16 #include <sys/stat.h>
17 #include <fcntl.h>
18 #include <string.h>
19 #include <dirent.h>
20 #include <time.h>
21 #include <pwd.h>
22 #include <grp.h>
23 #include <limits.h>
24 void child_process(int num, int n_actual) {
25     int pid;
26     if (n_actual == num)
27     {
28         srand( getpid() );
29         sleep(1);
30         for(int j = 0; j < n_actual; j++){
31             printf("\t");
32         }
33         printf("PPID = %i, PID = %i NIVEL = %i\n", getppid(), getpid(), n_actual);

```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

1: zsh + [] [] [] [] []

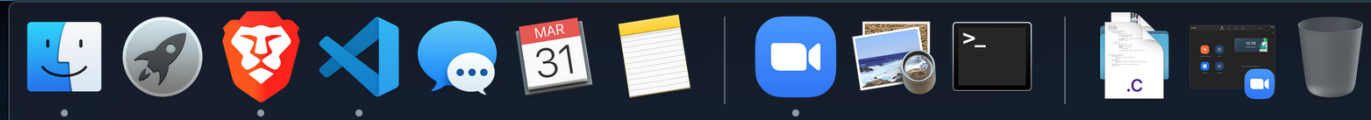
```
m4uz@MacBook-Pro-de-Irving act_colab_03 % ./main 3
PPID = 3476, PID = 3658 NIVEL = 0
  PPID = 3658, PID = 3659 NIVEL = 1
    PPID = 3659, PID = 3660 NIVEL = 2
      PPID = 3660, PID = 3661 NIVEL = 3
      PPID = 3660, PID = 3662 NIVEL = 3
      PPID = 3660, PID = 3666 NIVEL = 3
    PPID = 3659, PID = 3668 NIVEL = 2
      PPID = 3668, PID = 3669 NIVEL = 3
      PPID = 3668, PID = 3670 NIVEL = 3
      PPID = 3668, PID = 3671 NIVEL = 3
m4uz@MacBook-Pro-de-Irving act_colab_03 %
```



```
main.c test.sh
act_colab_03 > test.sh
1  #!/bin/bash
2
3  cal=10
4
5  gcc $1
6  if [[ $? -ne 0 ]] ; then
7      echo "grade: $cal"
8      exit 1
9  fi
10
11 ./a.out
12 if [[ $? -ne 254 ]] ; then
13     echo "grade: $cal"
14     exit 1
15 fi
16 cal=$((cal+10))
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 1: zsh

```
m4uz@MacBook-Pro-de-Irving act_colab_03 % ls
main  main.c  test.sh
m4uz@MacBook-Pro-de-Irving act_colab_03 % chmod +x test.sh
m4uz@MacBook-Pro-de-Irving act_colab_03 % ./test.sh main.c
usage: ./a.out number
./a.out: the parameter must be a positive integer number
./a.out: the parameter must be a positive integer number
./a.out: the parameter must be a positive integer number
PPID = 3751, PID = 3759 NIVEL = 0
  PPID = 3759, PID = 3760 NIVEL = 1
    PPID = 3760, PID = 3761 NIVEL = 2
      PPID = 3761, PID = 3762 NIVEL = 3
        PPID = 3761, PID = 3763 NIVEL = 3
          PPID = 3761, PID = 3775 NIVEL = 3
            PPID = 3760, PID = 3776 NIVEL = 2
              PPID = 3776, PID = 3777 NIVEL = 3
                PPID = 3776, PID = 3778 NIVEL = 3
                  PPID = 3776, PID = 3779 NIVEL = 3
=>grade: 100
m4uz@MacBook-Pro-de-Irving act_colab_03 %
```



Google

hora queretaro

Cerca de 8,790,000 resultados (0.52 segundos)

9:40
martes, 31 de marzo de 2020 (GMT-6)
Hora en Santiago de Querétaro, Qro.

www.worldtimeserver.com - hora-exacta-MX-QUE - ciudad de Querétaro, Querétaro, México - obtener la hora actual y la fecha exacta o hacer planes de viaje para un vuelo barato o hacer planes de viaje para un vuelo barato o hacer planes de viaje para un vuelo barato

www.worldtimeserver.com - hora-exacta-MX-QUE - ciudad de Querétaro, Querétaro, México - obtener la hora actual y la fecha exacta o hacer planes de viaje para un vuelo barato o hacer planes de viaje para un vuelo barato o hacer planes de viaje para un vuelo barato

time.is - Querétaro

La hora actual en Querétaro, México - Time and Date

www.zeitverschiebung.net - Querétaro - Querétaro

agprocesos.c - Visual Studio Code

```
43 }
44 }
45 }
46 }
47 bool isNumber(const char *str)
48 {
49     for (int i = 0; i < str.length(); i++)
50         if (!isdigit(str[i])) return false;
51     return true;
52 }
53 }
54 }
55 }
56 int main(int argc, char* argv[]) {
57     int pid, num, result, rld, i;
58
59     if (argc != 2) {
60         fprintf(stderr, "usage: %s number\n", argv[0]);
61         return -1;
62     }
63
64     num = atoi(argv[1]);
65     if (num < 1) {
66         fprintf(stderr, "%s: the parameter must be a positive integer number\n", argv[0]);
67         return -2;
68     }
69
70     child_process(num);
71     return 0;
72 }
73 }
```

Ln 58, Col 21 | Spaces: 4 | UTF-8 | C | Win32 | 09:41 a.m. 31/03/2020

google.com/search?q=hora queretaro

Cerca de 8,790,000 resultados

9:55

martes, 31 de marzo de 2020
Hora en Santiago de Chile

www.worldtimeserver.com • Hora local actual en Queretaro, Queretaro, México llamada telefónica o hacer planes de

www.worldtimeserver.com • Hora local actual en Queretaro, Mexico - obtener telefónica o hacer planes de

time.is • Queretaro

La hora actual en Queretaro
Hora exacta, zona horaria, día para Queretaro, México.

proceso.pdf

file | D:\Javier\20Mac\20Mendez\Documents\T...

de $NIVEL_2$ creará tres procesos de nivel $NIVEL_3$ y así sucesivamente $NIVEL_N$ creará N procesos de $NIVEL_N$. Cada proceso deberá desplegar información sobre el id de su padre y su propio id. Los procesos de nivel N dormirán 1 segundo antes de terminar. Cada proceso deberá esperar hasta que todos sus hijos hayan terminado.

Ejemplos de uso:

```

$ ./descending
usage: descending number

$ ./descending texto
descending: the parameter must be a positive integer number

$ ./descending texto
descending: the parameter must be a positive integer number

$ ./descending 12.14
descending: the parameter must be a positive integer number

$ ./descending -10
descending: the parameter must be a positive integer number

$ ./descending -10
descending: the parameter must be a positive integer number

$ ./descending 0
descending: the parameter must be a positive integer number

$ ./descending 3
PPID = 1234 PID = 1235 NIVEL = 0
    PPID = 1235 PID = 1236 NIVEL = 1
        PPID = 1236 PID = 1237 NIVEL = 2
            PPID = 1237 PID = 1238 NIVEL = 3
            PPID = 1237 PID = 1239 NIVEL = 3
            PPID = 1237 PID = 1240 NIVEL = 3
        PPID = 1236 PID = 1241 NIVEL = 2
        PPID = 1241 PID = 1242 NIVEL = 3
  
```

```

root@kali: ~/Desktop
desc.c:104:23: warning: passing argument 1 of 'isNumber' from incompatible pointer type [incompatible-pointer-types]
64 |     if (isNumber(argv[1])) {
    |         ~~~~~^~~~~
    |         char **
desc.c:148:23: note: expected 'char **' but argument is of type 'char *'
48 |     bool isNumber(char *s)
    |
    |
root@kali:~/Desktop rm -f desc.c
root@kali:~/Desktop nano desc.c
root@kali:~/Desktop gcc desc.c -o desc
root@kali:~/Desktop ./desc 12.2
usage: ./desc number
root@kali:~/Desktop rm -f desc.c
root@kali:~/Desktop nano desc.c
Show Applications gcc gcc desc.c -o desc
root@kali:~/Desktop# ./desc
  
```

```

File Edit Selection View Go Run ... apcprosool - Visual S...
C: apcprosool X
D:\Javier Mac Mendez\Documents\2019\semestre 2\programacion avanzada 2\p...
58     for (int i = 0; i < s[i]; i++)
59     {
60         if (!isdigit(s[i])) == false
61         {
62             return false;
63         }
64         return true;
65     }
66
67 int main(int argc, char* argv[]) {
68     int pid, num, result, rid, i;
69
70     if (argc != 2) {
71         fprintf(stderr, "usage: %s number\n", argv[0]);
72         return -1;
73     }
74     if (!isNumber(argv[1])) {
75         fprintf(stderr, "Error: the parameter must be a p...
76     }
  
```


Google

hora queretaro

Cerca de 8,790,000 resultados (0.73 segundos)

11:39

martes, 31 de marzo de 2020 (GMT-6)
Hora en Santiago de Querétaro, Qro.

www.worldtimeserver.com › hora-exacta-MX-QU
Hora local actual en Queretaro, Q
Queretaro, Querétaro, México - obtener la hora
llamada telefónica o hacer planes de viaje para

www.worldtimeserver.com › hora-exacta-MX-QU
Hora local actual en en Querétaro
Querétaro, México - obtener la hora actual y la
telefónica o hacer planes de viaje para un vuelo

time.is › Querétaro
La hora actual en Querétaro, Méx
Hora exacta, zona horaria, diferente zona horar
para Querétaro, México.

```
main.c
D: > Javier Mac Mendez > Documents > tec > sexto semestre > programacion avanzada > parcial 2 > C m
20 #include <time.h>
21 #include <pwd.h>
22 #include <grp.h>
23 #include <limits.h>
24 #include <ctype.h>
25 #include <stdbool.h>
26
27 void child_process(int num, int n_actual){
28     int pid;
29     if (n_actual == num)
30     {
31         srand( getpid() );
32         sleep(1);
33         for(int j = 0; j < n_actual; j++){
34             printf("\t");
35         }
36         printf("PPID = %i, PID = %i NIVEL = %i\n", getppid(), get
37         exit(0);
38     }
39     else
40     {
41         for(int j = 0; j < n_actual; j++){
42             printf("\t");
43         }
44         printf("PPID = %i, PID = %i NIVEL = %i\n", getppid(), get
45         for (int i = 0; i <= n_actual; i++)
46         {
47             if ( (pid = fork()) < 0 )
48             {
49                 perror("fork");
50                 exit(-4);
51             }
52             else if (pid == 0)
53             {
54                 n_actual++;
55                 child_process(num, n_actual);
56             }
57             else
58             {
59                 //
60             }
61         }
62     }
63 }
```

Ter... Mar 31 13:41

root@kali: ~/Desktop

```
PPID = 6037, PID = 6054 NIVEL = 2
PPID = 6054, PID = 6055 NIVEL = 3
PPID = 6055, PID = 6056 NIVEL =
PPID = 6055, PID = 6057 NIVEL =
PPID = 6055, PID = 6058 NIVEL =
PPID = 6055, PID = 6059 NIVEL =
PPID = 6054, PID = 6060 NIVEL = 3
PPID = 6060, PID = 6061 NIVEL =
PPID = 6060, PID = 6062 NIVEL =
PPID = 6060, PID = 6063 NIVEL =
PPID = 6060, PID = 6064 NIVEL =
PPID = 6054, PID = 6065 NIVEL = 3
PPID = 6065, PID = 6066 NIVEL =
PPID = 6065, PID = 6067 NIVEL =
PPID = 6065, PID = 6068 NIVEL =
PPID = 6065, PID = 6069 NIVEL =
```

desktop#

CTRL DERECHA

sqldeveloper