Set_G_Lab4

Question 1 (1 point) ✓ *Saved*

First compile your code (https://cocalc.com/) to obtain main1.out and then run main1.out using:

./main1.out & pstree -pT | grep main1.out

main1.c

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

int main(){

    pid_t fork_return = fork();
    pid_t  pid = getpid();
    printf("fork return value = %d , pid = %d \n", fork_return, pid);
    sleep(1);

    return 0;
}
```

Answer the following questions (Q2, Q3, Q4, Q5):

Insert a screenshot of your program's output after execution.

Set_G_Lab4

Insert a screenshot of your program's output after execution.

Paragraph

Add a File | Record Audio | Record Video

Screenshot from 2026-02-04 11-02-14.png (114.43 KB)

Question 2 (1 point) ✓ *Saved*

What is the main purpose of the **`getpid()`** function?

- ○ To get the thread ID
- ○ To get the process priority
- ● To get the process ID of the current process
- ○ To get the parent process ID

Set_G_Lab4

Question 3 (1 point) ✓ *Saved*

What type of value does getpid() return?

- ○ A floating-point number
- ○ A memory address
- ○ A string containing the process name
- ● An integer representing the process ID

Question 4 (1 point) ✓ *Saved*

Refer to your results from **Question 1** and select the correct option describing the parent process ($\mathbf{PID_P}$) and child process ($\mathbf{PID_C}$).

- ● $PID_P > 0$ and $PID_C = 0$
- ○ $PID_P = 0$ and $PID_C > 0$
- ○ $PID_P \neq PID_C$
- ○ $PID_P = PID_C$

Set_G_Lab4

Question 5 (2 points) ✓ *Saved*

Based on the results of Question 1, choose the correct fork() return value for both the parent ($\mathbf{PID_P}$) and child ($\mathbf{PID_C}$) processes

3 In the parent process, the return value of fork() is

1 In the child process, the return value of fork() is

1. 0
2. PID_P
3. PID_C
4. -1

Set_G_Lab4

100% − + Reset

Question 6 (1 point) ✓ *Saved*

Consider the following C code, first compile your code to obtain the binary file main2.out (use https://cocalc.com/) and then run it.

gcc -static main2.c -o main2.out

main2.c

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

int global_variable = 100;
int un_global_variable;

int main()
{
    char str[30];
    int *ptr;
    int local_variable = 200;
    int un_local_variable;

    ptr = (int*) malloc(2 * sizeof(int));

    pid_t pid;
    pid = getpid();

    printf("\n");
    printf(" Global variable                : %p\n", &global_variable);
    printf(" Uninitialized global variable  : %p\n", &un_global_variable);

    printf(" Local variable                 : %p\n", &local_variable);
    printf(" Uninitialized local variable   : %p\n", &un_local_variable);

    printf(" Memory allocation              : %p\n\n", ptr);
    printf("  ======== The Process Address Space ( pid = %d )=======\n\n", pid);

    sprintf(str, "cat -b /proc/%d/maps", pid);
    system(str);
    free(ptr);

    return 0;
}
```

Set_G_Lab4

Paragraph

```
~$ ./main2.out

Global variable                : 0x4ac0d0
Uninitialized global variable  : 0x4adb30

Local variable                 : 0x7ffd5c0ea4dc
Uninitialized local variable   : 0x7ffd5c0ea4e0

Memory allocation              : 0x1d7ed900

======== The Process Address Space ( pid = 455 )========

    1  00400000-00401000 r--p 00000000 00:220 7                                 /home/user/main2.out
    2  00401000-00481000 r-xp 00001000 00:220 7                                 /home/user/main2.out
    3  00481000-004a7000 r--p 00081000 00:220 7                                 /home/user/main2.out
    4  004a7000-004ac000 r--p 000a6000 00:220 7                                 /home/user/main2.out
    5  004ac000-004ae000 rw-p 000ab000 00:220 7                                 /home/user/main2.out
    6  004ae000-004b4000 rw-p 00000000 00:00 0
    7  1d7ec000-1d80e000 rw-p 00000000 00:00 0                                  [heap]
    8  7ffd5c0ca000-7ffd5c0ec000 rw-p 00000000 00:00 0                          [stack]
    9  7ffd5c183000-7ffd5c187000 r--p 00000000 00:00 0                          [vvar]
```

New Log Find Servers Users Upgrades Processes

Add a File | Record Audio | Record Video

Screenshot from 2026-02-04 11-08-48.png (120.16 KB)

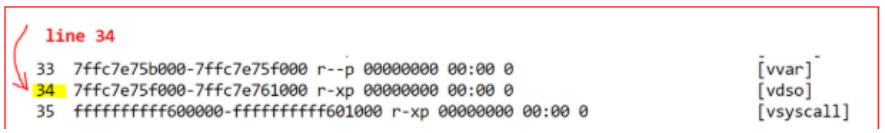
Question 7 (3 points) ✓ *Saved*

Refer to your results from **Question 6** and obtain the location of the initialized and uninitialized global variables (the line number)? e.g. 12 [5]

Refer to your results from **Question 6** and obtain the location of the initialized and uninitialized local variables (the line number)? e.g. 12 [8]

Refer to your results from **Question 6** and obtain the location of allocated memory (the line number)? e.g. 12 [7]

e.g.



Set_G_Lab4

Question 8 (2 points) ✓ *Saved*

A computer system can hold three identical processes in its main memory at a time. Each process has an independent probability p = 80%.

Calculate the overall CPU utilization, defined as the probability that at least one process is ready to use the CPU. e.g. 0.123 0.992

Now consider only the scenario in which exactly one process is waiting for I/O, and the other two are ready. What is the CPU utilization in this specific case? e.g. 0.123

0.384

Question 9 (1 point) ✓ *Saved*

Consider the following c code, compile your code to obtain the binary file main3.out.

gcc main3.c -o main3.out

main3.c

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

int main() {
    pid_t pid = fork();

    if (pid < 0) {
        perror("fork failed");
        exit(1);
    }

    if (pid == 0) {
```

Set_G_Lab4

Page 1:

1 ✓ 2 ✓ 3 ✓
4 ✓ 5 ✓ 6 ✓
7 ✓ 8 ✓ 9 ✓
10 ✓ 11 ✓

Quiz Information

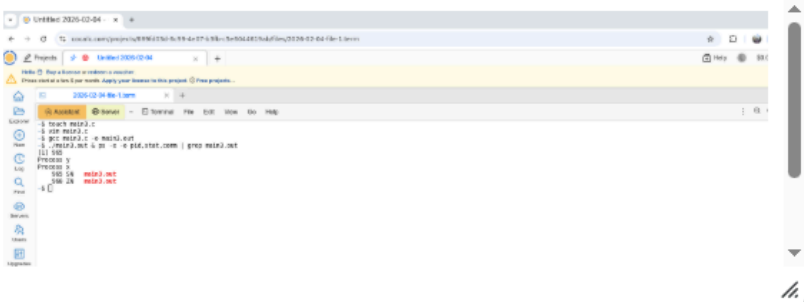
```
12
13      if (pid == 0) {
14          printf("Process x \n");
15          exit(0);
16      } else {
17          printf("Process y \n");
18          sleep(30);
19      }
20
21      return 0;
22  }
```

Run your program and use its output to answer the following questions:

./main3.out & ps -e -o pid,stat,comm | grep main3.out

Insert a screenshot of your program's output.

Paragraph



Add a File | Record Audio | Record Video

Question 10 (2 points) ✓ *Saved*

Refer to your results from **Question 9**, the child process is e.g. x or y

x

Refer to your results from **Question 9**, the parent process is e.g. x or y

y

Question 11 (2 points) ✓ *Saved*

Question 11 (2 points) ✓ *Saved*

Refer to your results from **Question 9**, Specify the process state of both the child and the parent.

2 Process state of Parent

4 Process state of Child

1. T
2. S
3. D
4. Z
5. R