

Iniciado em	terça, 5 jul 2022, 20:22
Estado	Finalizada
Concluída em	quinta, 7 jul 2022, 15:25
Tempo empregado	1 dia 19 horas
Avaliar	Ainda não avaliado

[Atendimento](#)

Questão 1

Completo

Vale 1,00 ponto(s).

Crie um código em C para gerar três processos-filho usando o `fork()`.

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

int main(int argc, char **argv){
    pid_t pid_filho[3];
    pid_t pid_pai = getpid();
    int i;

    printf("PID do pai: %d \n \n",pid_pai);

    for ( i = 0; i < 3; i++)
    {
        if (getpid()==pid_pai)
        {
            pid_filho[i] = fork();
            printf("PID do pai: %d \nPID do filho: %d\n", getpid(), pid_filho[i]);
        }
    }
}
```

Atendimento

}

 [main.c](#)[Atendimento](#)

Questão **2**

Completo

Vale 1,00 ponto(s).

Crie um código em C que recebe o nome de diversos comandos pelos argumentos de entrada (**argc** e ***argv[]**), e executa cada um sequencialmente usando **system()**. Por exemplo, considerando que o código criado recebeu o nome de 'serial_system', e que ele foi executado em março de 2017 na pasta '/Sistemas_Embarcados/Code/06_Processos', que contem diversos arquivos:

```
$ ./serial_system pwd echo ls echo cal
~/Sistemas_Embarcados/Code/06_Processos

Ex1.c    Ex3a.c    Ex4.c    Ex6.c    Ex8.c    serial_system
Ex2.c    Ex3b.c    Ex5.c    Ex7.c    Ex9.c

    Março 2017
Do Se Te Qu Qu Se Sá
      1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

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

```
int main(int argc, char **argv){
    int i;

    for ( i = 1; i < argc; i++)
    {
```

Atendimento

```
    system(argv[i]);  
}  
  
return 0;  
}  
↩ main.c
```

[Atendimento](#)

Questão **3**

Completo

Vale 1,00 ponto(s).

Crie um código em C que recebe o nome de diversos comandos pelos argumentos de entrada (`argc` e `*argv[]`), e executa cada um usando `fork()` e `exec()`

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

int main(int argc, char **argv){
    int i;
    pid_t pid_pai = getpid();
    pid_t pid_filho;
    char *argumentos[] = {NULL, NULL};

    for ( i = 1; i < argc; i++)
    {
        pid_filho = fork();
        if (pid_filho==0)
        {
            argumentos[0] = argv[i];
            execvp(argumentos[0], argumentos);
        }
        else{
            wait(NULL);
        }
    }

    return 0;
}
```

 [main.c](#)

Atendimento

Atendimento

Questão 4

Completo

Vale 1,00 ponto(s).

Crie um código em C que gera três processos-filho usando o `fork()`, e que cada processo-filho chama a seguinte função uma vez:

```
int v_global = 0; // Variavel global para este exemplo
void Incrementa_Variavel_Global(pid_t id_atual)
{
    v_global++;
    printf("ID do processo que executou esta funcao = %d\n", id_atual);
    printf("v_global = %d\n", v_global);
}
```

(Repare que a função `Incrementa_Variavel_Global()` recebe como entrada o ID do processo que a chamou.) Responda: a variável global `v_global` foi compartilhada por todos os processos-filho, ou cada processo enxergou um valor diferente para esta variável?

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

```
int v_global = 0; // Variavel global para este exemplo
void Incrementa_Variavel_Global(pid_t id_atual)
{
    v_global++;
    printf("ID do processo que executou esta funcao = %d\n", id_atual);
    printf("v_global = %d\n", v_global);
}
```

```
int main(void){
```

Atendimento


```
pid_t pid_pai = getpid();
pid_t pid_filho[3];
int i;

for ( i = 0; i < 3; i++)
{
    if (getpid()==pid_pai)
    {
        pid_filho[i] = fork();
        if(pid_filho[i]==0){
            Incrementa_Variavel_Global(getpid());
        }
        else{
            wait(NULL);
        }
    }
}

return 0;
}
```

 [main.c](#)



Atendimento