

Iniciado em	quinta, 7 jul 2022, 17:19
Estado	Finalizada
Concluída em	quinta, 7 jul 2022, 20:56
Tempo empregado	3 horas 37 minutos
Avaliar	Ainda não avaliado

Questão 1

Correto

Atingiu 1,00 de 1,00

Quantos pipes serão criados após as linhas de código a seguir?

```
int pid;  
int fd[2];  
pipe(fd);  
pid = fork();
```

Escolha uma opção:

- ☒ a. 1
- ☐ b. 2
- ☐ c. 3
- ☐ d. 4
- ☐ e. 5



Sua resposta está correta.

A resposta correta é: 1

Atendimento

Questão 2

Correto

Atingiu 1,00 de 1,00

Quantos pipes serão criados após as linhas de código a seguir?

```
int pid;  
int fd[2];  
pid = fork();  
pipe(fd);
```

Escolha uma opção:

- ☐ a. 1
- ☒ b. 2
- ☐ c. 3
- ☐ d. 4
- ☐ e. 5



Sua resposta está correta.

A resposta correta é: 2

[Atendimento](#)

Questão 3

Completo

Vale 1,00 ponto(s).

Considere o código a seguir:

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

void tratamento_alarme(int sig)
{
    system("date +%H:%M:%S.%N");
    alarm(1);
}

int main()
{
    signal(SIGALRM, tratamento_alarme);
    alarm(1);
    printf("Aperte CTRL+C para acabar:\n");
    while(1);
    return 0;
}
```

Sabendo que a função `alarm()` tem como entrada a quantidade de segundos para terminar a contagem, quão precisos são os alarmes criados neste código? De onde vem a imprecisão? Este é um método confiável para desenvolver aplicações em tempo real?

Output:

Os alarmes criados nesse código são precisos com relação aos segundos, verificados pela imagem acima. Porém, quando avançamos na escala, notamos imprecisões nos milissegundos. Essa imprecisão ocorre pois o programa, além de cronometrar os 1 segundo, executa outros tipos de instrução, causando a imprecisão. Além de não trabalharmos com sistemas ideais. Sendo assim, esse método deve ser analisado para a aplicação em tempo real, se caso a aplicação for apenas acender uma luz remotamente não teria problema, mas se a aplicação for o controle de uma máquina, esse atraso fará muita diferença.

Atendimento



Atendimento

Questão **4**

Completo

Vale 1,00 ponto(s).

Crie um programa em C que cria um processo-filho e um pipe de comunicação. Faça com que o processo-pai envie os valores 1, 2, 3, 4, 5, 6, 7, 8, 9 e 10 para o processo-filho, com intervalos de 1 segundo entre cada envio. Depois de o processo-pai enviar o número 10, ele aguarda 1 segundo e termina a execução. O processo-filho escreve na tela cada valor recebido, e quando ele receber o valor 10, ele termina a execução.

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

int main(void){
    pid_t pid_pai = getpid();
    pid_t pid_filho;
    int i,j;
    int fd[2];
    int lido = 0;

    pipe(fd);
    pid_filho = fork();
    if(pid_filho == 0){
        while (lido < 10)
        {
            read(fd[0],&lido, sizeof(lido));
```

Atendimento

```
        printf("%d\n",lido);
    }

}

else{
    for ( i = 1; i < 11; i++)
    {
        write(fd[1],&i,sizeof(i));
        sleep(1);

    }
    sleep(1);
}

return 0;
}

↔ main.c
```

Atendimento

Questão **5**

Completo

Vale 1,00 ponto(s).

Crie um programa em C que cria um processo-filho e um pipe de comunicação. Utilize o pipe para executar a seguinte conversa entre processos:

```
FILHO: Pai, qual é a verdadeira essência da sabedoria?  
PAI: Não façais nada violento, praticai somente aquilo que é justo e equilibrado.  
FILHO: Mas até uma criança de três anos sabe disso!  
PAI: Sim, mas é uma coisa difícil de ser praticada até mesmo por um velho como eu...
```

Neste exercício, quem recebe a mensagem via pipe é quem as escreve no terminal.

```
#include <stdio.h>  
#include <stdlib.h>  
#include <unistd.h>  
#include <string.h>  
#include <sys/wait.h>  
  
int main(void){  
    pid_t pid_pai = getpid();  
    pid_t pid_filho;  
    int i,j;  
    int fd[2],fp[2];  
    char msg_in[100],msg_out[100];  
  
    pipe(fd);  
    pipe(fp);  
    pid_filho = fork();  
    if(pid_filho == 0){
```

Atendimento

```
strcpy(msg_out,"FILHO: Pai, qual é a verdadeira essência da sabedoria?");
write(fd[1],msg_out,100);
read(fp[0],msg_in,100);
printf("%s\n",msg_in);
strcpy(msg_out, "FILHO: Mas até uma criança de três anos sabe disso!");
write(fd[1],msg_out,100);
read(fp[0],msg_in,100);
printf("%s\n",msg_in);

}
else{
    read(fd[0],msg_in,100);
    printf("%s\n",msg_in);
    strcpy(msg_out,"PAI: Não façais nada violento, praticai somente aquilo que é justo e equilibrado.");
    write(fp[1],msg_out,100);
    read(fd[0],msg_in,100);
    printf("%s\n",msg_in);
    strcpy(msg_out,"PAI: Sim, mas é uma coisa difícil de ser praticada até mesmo por um velho como eu...");
    write(fp[1],msg_out,100);
    wait(NULL);
}

return 0;
}
```

 [_main.c](#)

Atendimento

Questão 6

Completo

Vale 1,00 ponto(s).

Crie um programa em C que cria dois processos-filhos e um pipe de comunicação. Utilize o pipe para executar a seguinte conversa entre processos:

FILH01: Quando o vento passa, é a bandeira que se move.

FILH02: Não, é o vento que se move.

PAI: Os dois se enganam. É a mente dos senhores que se move.

Neste exercício, quem recebe a mensagem via pipe é quem as escreve no terminal.

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

```
int fim_espera=0;
```

```
void tratamento_SIGUSR1()
{
    fim_espera=1;
}
```

```
int main(void){
    pid_t pid_pai = getpid();
    pid_t pid_filho[2];
```

Atendimento

```
int i,j;
int fd[2],fp[2];
char msg_in[100],msg_out[100];
signal(SIGUSR1, tratamento_SIGUSR1);

pipe(fd);
pipe(fp);
pid_filho[0] = fork();
if(getpid() == pid_pai){
    pid_filho[1] = fork();
}
if(pid_filho[0] == 0){
    strcpy(msg_out,"FILHO1: Quando o vento passa, é a bandeira que se move.");
    write(fd[1],msg_out,100);
    fim_espera = 0;
    while(fim_espera == 0);
    read(fd[0],msg_in,100);
    printf("%s\n",msg_in);
}
else if (pid_filho[1] == 0)
{
    strcpy(msg_out,"FILHO2: Não, é o vento que se move.");
    write(fp[1],msg_out,100);
}
else if (getpid() == pid_pai){
    read(fd[0],msg_in,100);
    printf("%s\n",msg_in);
    read(fp[0],msg_in,100);
```

```
printf("%s\n",msg_in);
strcpy(msg_out,"PAI: Sim, mas é uma coisa difícil de ser praticada até mesmo por um velho como eu...");
write(fd[1],msg_out,100);
kill(pid_filho[0],SIGUSR1);
wait(NULL);
}
```

```
return 0;
```

```
}
```

 [main.c](#)



Atendimento

Atendimento