

1. Aqui está um programa em C que cria dois processos filhos para gravar números de 1 a 10 e letras de A a Z em arquivos diferentes. O processo pai lista o conteúdo dos arquivos após os filhos terminarem:

```
``c

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/wait.h>

#include <fcntl.h>

void writeNumbers() {

    FILE *file = fopen("numbers.txt", "w");

    if (file != NULL) {

        for (int i = 1; i <= 10; i++) {

            fprintf(file, "%d ", i);

        }

        fclose(file);

    }

}

void writeLetters() {

    FILE *file = fopen("letters.txt", "w");

    if (file != NULL) {

        for (char c = 'A'; c <= 'Z'; c++) {

            fprintf(file, "%c ", c);

        }

        fclose(file);

    }

}
```

```
int main() {

    pid_t child1, child2;

    child1 = fork();

    if (child1 < 0) {

        perror("Erro ao criar o primeiro filho");

        exit(1);

    } else if (child1 == 0) {

        // Primeiro filho escreve números

        writeNumbers();

        exit(0);

    } else {

        // Processo pai

        child2 = fork();

        if (child2 < 0) {

            perror("Erro ao criar o segundo filho");

            exit(1);

        } else if (child2 == 0) {

            // Segundo filho escreve letras

            writeLetters();

            exit(0);

        } else {

            // Processo pai aguarda os filhos terminarem

            wait(NULL);

            wait(NULL);

            // Lista o conteúdo dos arquivos

            printf("Conteúdo de numbers.txt:\n");

            system("cat numbers.txt");

            printf("\nConteúdo de letters.txt:\n");
```

```

        system("cat letters.txt");
    }
}

return 0;
}
...

```

Certifique-se de que os arquivos "numbers.txt" e "letters.txt" não existam antes de executar o programa, ou os novos conteúdos serão adicionados aos arquivos existentes.

2. Abaixo está um programa em C que permite ao usuário remover um arquivo existente. O caminho do arquivo é passado como argumento na linha de comando:

```

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

int main(int argc, char *argv[]) {
 if (argc != 2) {
 printf("Uso: %s <caminho_do_arquivo>\n", argv[0]);
 exit(1);
 }

 char *filename = argv[1];

 if (remove(filename) == 0) {
 printf("O arquivo '%s' foi removido com sucesso.\n", filename);
 } else {
 perror("Erro ao remover o arquivo");
 exit(1);
 }
}

```

```
}

 return 0;
}
'''
```

O programa recebe o caminho do arquivo como argumento e tenta removê-lo. Certifique-se de ter as permissões adequadas para excluir o arquivo.