**GDB For Beginner\_Assignments\_V0.2**

1. gdb\_practice\_array.c contains a code with bugs. The actual output is supposed to generate a sorted list and then search for 3 strings in that as shown in actual\_output. Find out the bugs in practise.c using gdb.

**Ans**: The issue in the provided code is related to how the variable i is being handled in the main() function. Specifically, i is not initialized before being used in the while loop that prints the initial list of critters and after sorting. Because i is uninitialized, its value is indeterminate, which can lead to unexpected behavior during the loop execution.

Here's a step-by-step breakdown of the issues and the solution:

1. Uninitialized variable i in main

The variable i is used to iterate through the muppets array and print the critters, but it is never initialized. This leads to undefined behavior as i starts with some random value.

The following lines:

while (i < count)

{

print\_critter (&muppets[i]);

i++;

}

The variable i should be initialized to 0 before it is used:

int i = 0; // Initialize i to 0

while (i < count)

{

print\_critter (&muppets[i]);

i++;

}

2. After sorting, i is not reset

After the sorting operation with qsort1(), the program attempts to print the sorted list again. However, since i was not reset, it continues from its last value, which was likely greater than count after the first loop. This leads to no critters being printed in the second loop. You need to reset i to 0 before the second loop:

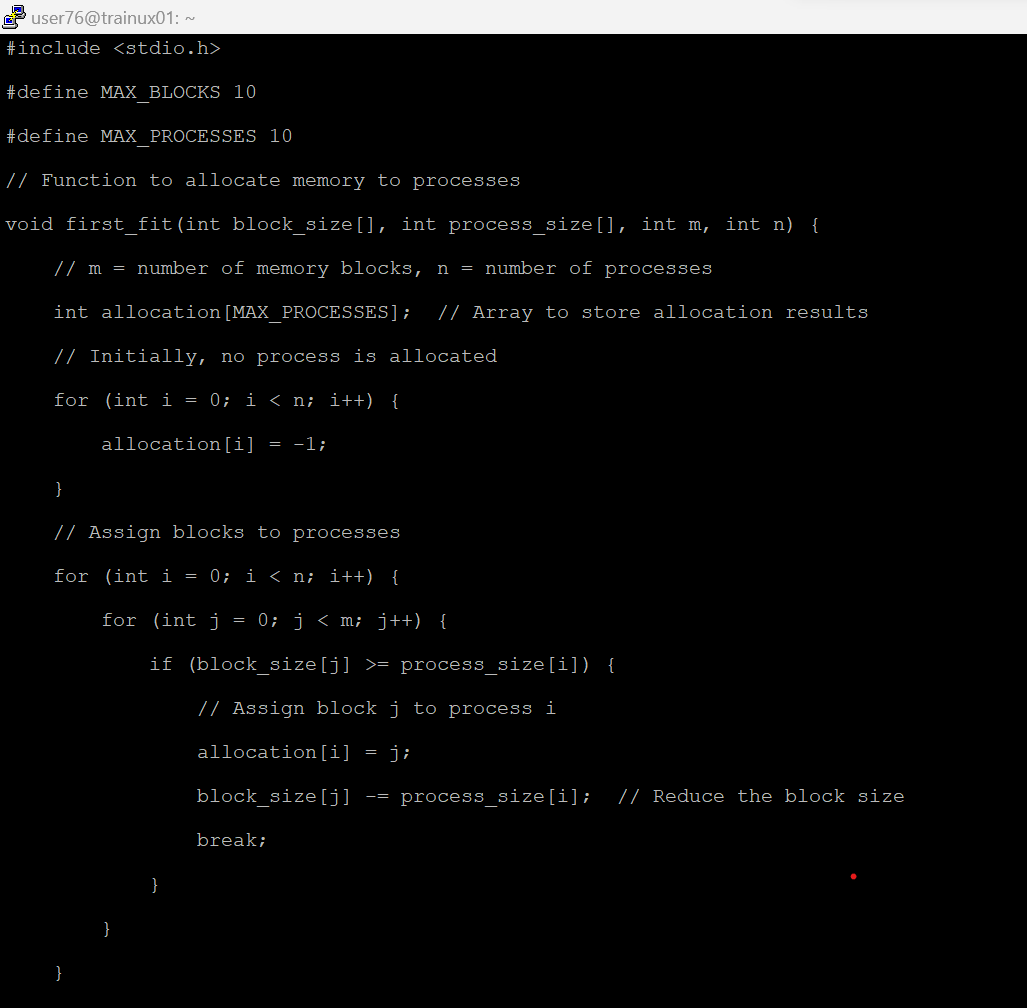
i = 0; // Reset i to 0 before printing the sorted list

while (i < count)

{

print\_critter (&muppets[i]); i++;}

2. First Fit Memory Management algorithm is implemented in the file first\_fit\_bugs.c which contains a few bugs. Remove these bugs with the help of gdb so that it gives expected output. Refer link below for algorithm details.



A screenshot of a computer program

Description automatically generated

