Task 1K: Using #pragma pack to Observe Structure Memory Layout using GDB

Objective:
To understand how structure padding and alignment work in C and how the #pragma pack directive affects the memory layout of a structure. This task uses GDB to observe structure size, member addresses, and memory layout with and without packing.

1. Concept Overview

By default, the compiler adds padding between structure members to satisfy alignment requirements for performance reasons. The #pragma pack directive is used to change this alignment, thereby reducing memory usage at the cost of performance.

2. Program Code

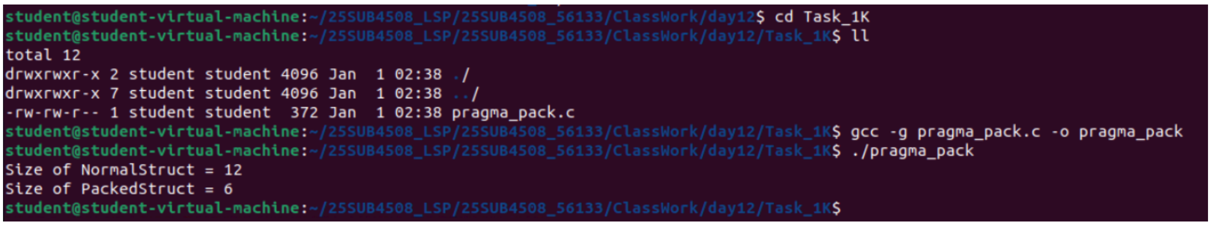
```
#include <stdio.h>

 struct NormalStruct {
         char a;
         int b;
         char c;
 };

 #pragma pack(1)
 struct PackedStruct {
         char a;
         int b;
         char c;
 };
 #pragma pack()

 int main() {
         struct NormalStruct ns;
         struct PackedStruct ps;

         printf("Size of NormalStruct = %lu\n", sizeof(ns));
         printf("Size of PackedStruct = %lu\n", sizeof(ps));
         return 0;
 }
```

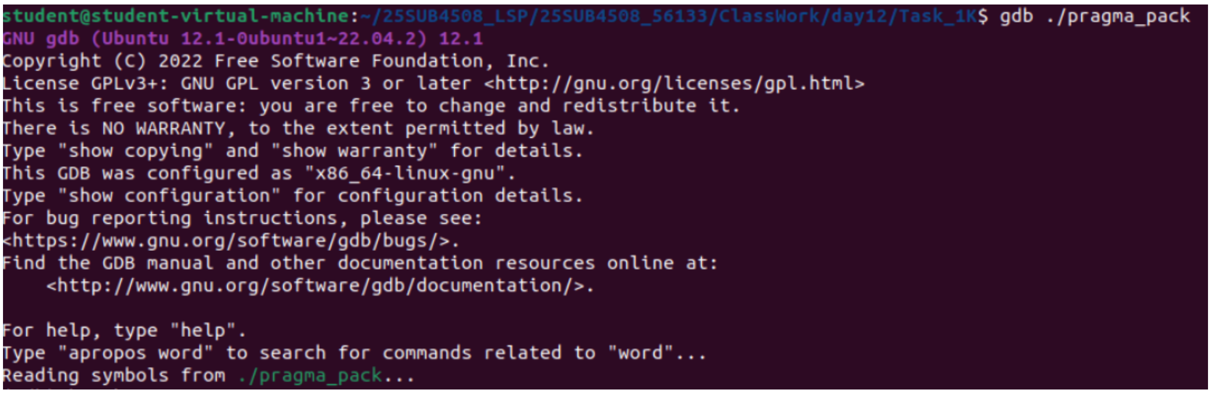


```
student@student-virtual-machine:~/25SUB4508_LSP/25SUB4508_56133/ClassWork/day12$ cd Task_1K
student@student-virtual-machine:~/25SUB4508_LSP/25SUB4508_56133/ClassWork/day12/Task_1K$ ll
total 12
drwxrwxr-x 2 student student 4096 Jan  1 02:38 ./
drwxrwxr-x 7 student student 4096 Jan  1 02:38 ../
-rw-rw-r-- 1 student student  372 Jan  1 02:38 pragma_pack.c
student@student-virtual-machine:~/25SUB4508_LSP/25SUB4508_56133/ClassWork/day12/Task_1K$ gcc -g pragma_pack.c -o pragma_pack
student@student-virtual-machine:~/25SUB4508_LSP/25SUB4508_56133/ClassWork/day12/Task_1K$ ./pragma_pack
Size of NormalStruct = 12
Size of PackedStruct = 6
student@student-virtual-machine:~/25SUB4508_LSP/25SUB4508_56133/ClassWork/day12/Task_1K$
```

3. Compilation Instructions

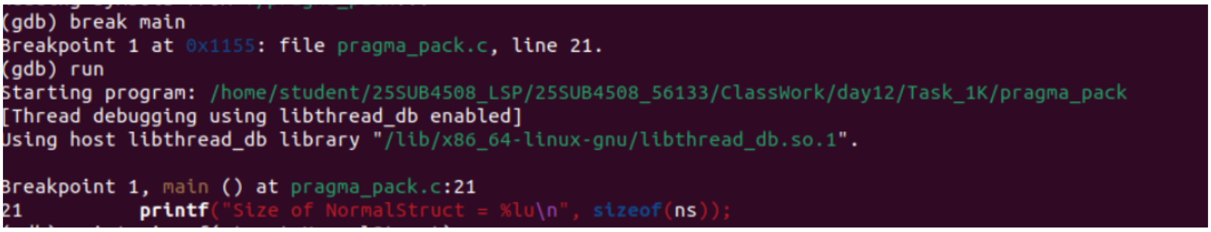
Compile the program with debugging symbols enabled:

gcc -g pragma_pack.c -o pragma_pack



```
student@student-virtual-machine:~/25SUB4508_LSP/25SUB4508_56133/ClassWork/day12/Task_1K$ gdb ./pragma_pack
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04.2) 12.1
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./pragma_pack...
```



```
(gdb) break main
Breakpoint 1 at 0x1155: file pragma_pack.c, line 21.
(gdb) run
Starting program: /home/student/25SUB4508_LSP/25SUB4508_56133/ClassWork/day12/Task_1K/pragma_pack
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at pragma_pack.c:21
21          printf("Size of NormalStruct = %lu\n", sizeof(ns));
```

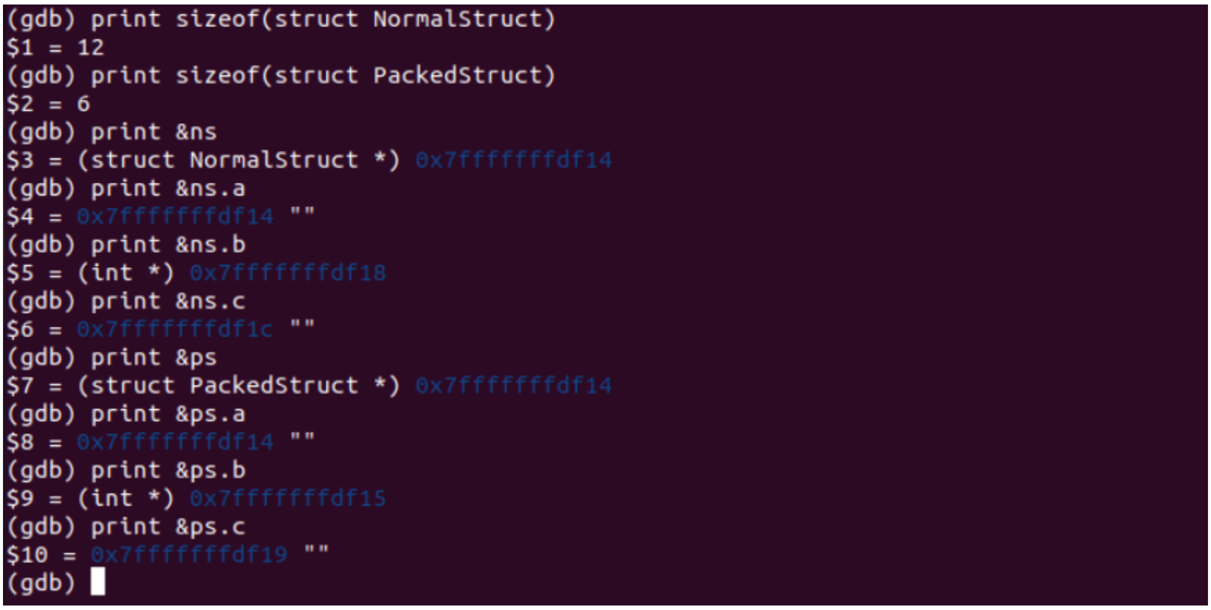
4. GDB Commands Used

```
break main
 run

 print sizeof(struct NormalStruct)
 print sizeof(struct PackedStruct)

 print &ns
 print &ns.a
 print &ns.b
 print &ns.c

 print &ps
 print &ps.a
 print &ps.b
 print &ps.c
```



```
(gdb) print sizeof(struct NormalStruct)
$1 = 12
(gdb) print sizeof(struct PackedStruct)
$2 = 6
(gdb) print &ns
$3 = (struct NormalStruct *) 0x7fffffffdf14
(gdb) print &ns.a
$4 = 0x7fffffffdf14 ""
(gdb) print &ns.b
$5 = (int *) 0x7fffffffdf18
(gdb) print &ns.c
$6 = 0x7fffffffdf1c ""
(gdb) print &ps
$7 = (struct PackedStruct *) 0x7fffffffdf14
(gdb) print &ps.a
$8 = 0x7fffffffdf14 ""
(gdb) print &ps.b
$9 = (int *) 0x7fffffffdf15
(gdb) print &ps.c
$10 = 0x7fffffffdf19 ""
(gdb) 
```

5. Observations & Explanation

1. NormalStruct contains padding bytes added by the compiler.
2. PackedStruct removes padding and aligns members at 1-byte boundaries.
3. Memory addresses of PackedStruct members are contiguous.
4. Size reduction is achieved using #pragma pack(1).

6. Memory Layout Comparison

NormalStruct:
| char | padding | int | char | padding |

PackedStruct:
| char | int | char |

7. Conclusion

This task demonstrates how #pragma pack alters structure alignment and memory layout. It is useful in low-level programming, embedded systems, and network protocol implementations where memory size is critical.