Unix Assignment10

Team19 108072244 邱煒甯 108032053 陳凱揚 112065525 簡佩如

q1

Firstly, `pthread_once` is used with the `init_done` variable to ensure that function `thread_init` is executed exactly once. This is critical for setting up thread-specific data without race conditions. Also in `thread_init`, this function creates a key for thread-specific data. Each thread can associate a value with this key using `pthread_setspecific`, and the value can be retrieved with `pthread_getspecific`, which allows each thread to have its own env string buffer without collision between threads. What's more, the `env_mutex` mutex is used to protect the shared environment (`environ`) from concurrent access by multiple threads. All of the above techniques help the function to be thread-safe.

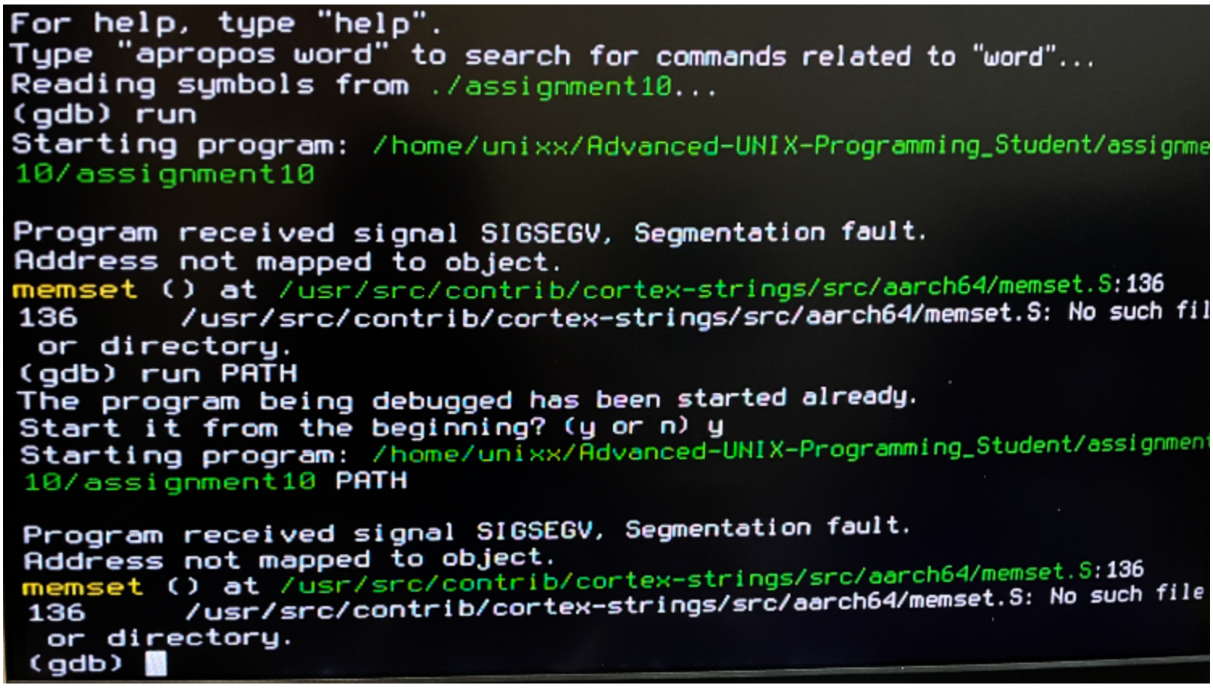
q2

While blocking signals can prevent signal handlers from interrupting the function, it does not make the function itself async-signal-safe. This is because the function may call other functions that are not async-signal-safe (such as `malloc`, `free`, and `pthread_mutex_lock/unlock`). Therefore, just blocking signals is not sufficient to make this function async-signal safe.

q3

If we run assignment10.c directly on FreeBSD it will crash. Because the `getenv` function originally exists in FreeBSD, the `getenv` function name in assignment10.c may conflict with the `getenv` in the standard function library (`stdlib.h`), causing the compiler to be unable to recognize our own defined `getenv` function.

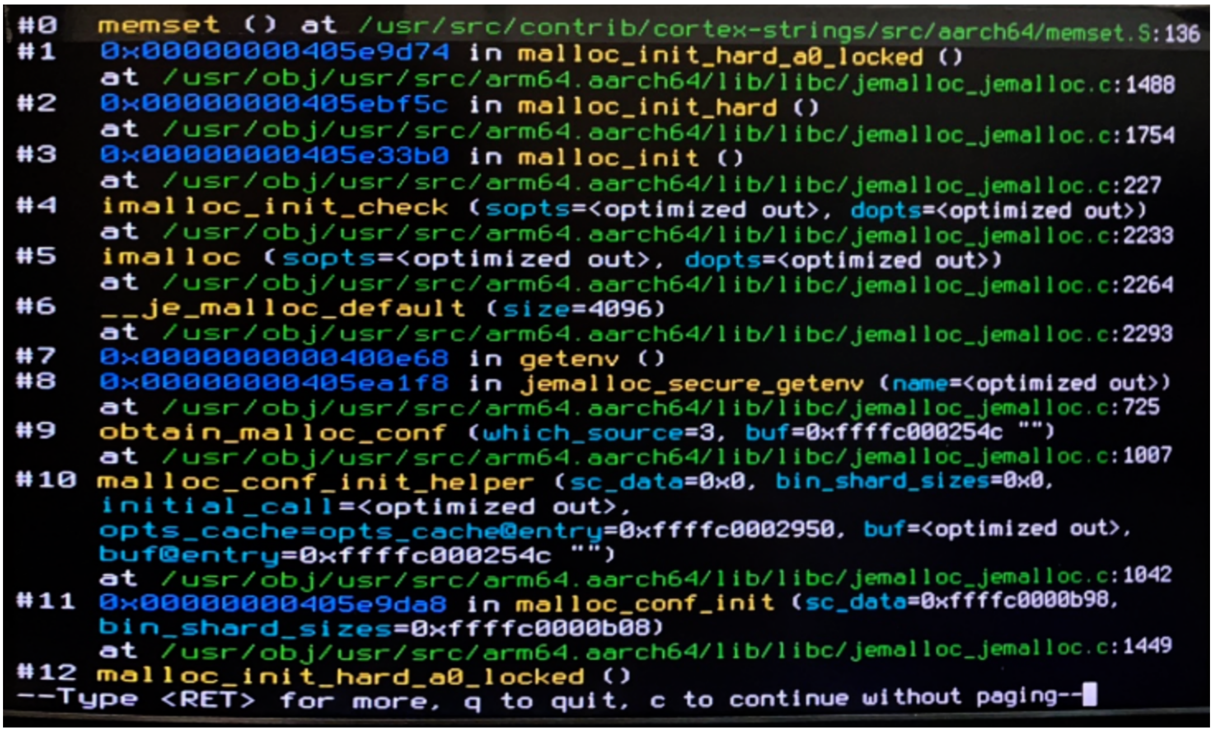
When using `gdb`, we find that it stops during the `getenv` function (output of gdb in the following images). It can be executed correctly if we change the name of `getenv` in assignment10.c (e.g., `get_env`).



```
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./assignment10...
(gdb) run
Starting program: /home/unixx/Advanced-UNIX-Programming_Student/assignme
10/assignment10

Program received signal SIGSEGV, Segmentation fault.
Address not mapped to object.
memset () at /usr/src/contrib/cortex-strings/src/aarch64/memset.S:136
136         /usr/src/contrib/cortex-strings/src/aarch64/memset.S: No such fil
 or directory.
(gdb) run PATH
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/unixx/Advanced-UNIX-Programming_Student/assignment
10/assignment10 PATH

Program received signal SIGSEGV, Segmentation fault.
Address not mapped to object.
memset () at /usr/src/contrib/cortex-strings/src/aarch64/memset.S:136
136         /usr/src/contrib/cortex-strings/src/aarch64/memset.S: No such file
 or directory.
(gdb)
```



```
#0  memset () at /usr/src/contrib/cortex-strings/src/aarch64/memset.S:136
#1  0x00000000405e9d74 in malloc_init_hard_a0_locked ()
    at /usr/obj/usr/src/arm64.aarch64/lib/libc/jemalloc_jemalloc.c:1488
#2  0x00000000405ebf5c in malloc_init_hard ()
    at /usr/obj/usr/src/arm64.aarch64/lib/libc/jemalloc_jemalloc.c:1754
#3  0x00000000405e33b0 in malloc_init ()
    at /usr/obj/usr/src/arm64.aarch64/lib/libc/jemalloc_jemalloc.c:227
#4  imalloc_init_check (sopts=<optimized out>, dopts=<optimized out>)
    at /usr/obj/usr/src/arm64.aarch64/lib/libc/jemalloc_jemalloc.c:2233
#5  imalloc (sopts=<optimized out>, dopts=<optimized out>)
    at /usr/obj/usr/src/arm64.aarch64/lib/libc/jemalloc_jemalloc.c:2264
#6  __je_malloc_default (size=4096)
    at /usr/obj/usr/src/arm64.aarch64/lib/libc/jemalloc_jemalloc.c:2293
#7  0x0000000000400e68 in getenv ()
#8  0x00000000405ea1f8 in jemalloc_secure_getenv (name=<optimized out>)
    at /usr/obj/usr/src/arm64.aarch64/lib/libc/jemalloc_jemalloc.c:725
#9  obtain_malloc_conf (which_source=3, buf=0xffffc000254c "")
    at /usr/obj/usr/src/arm64.aarch64/lib/libc/jemalloc_jemalloc.c:1007
#10 malloc_conf_init_helper (sc_data=0x0, bin_shard_sizes=0x0,
    initial_call=<optimized out>,
    opts_cache=opts_cache@entry=0xffffc0002950, buf=<optimized out>,
    buf@entry=0xffffc000254c "")
    at /usr/obj/usr/src/arm64.aarch64/lib/libc/jemalloc_jemalloc.c:1042
#11 0x00000000405e9da8 in malloc_conf_init (sc_data=0xffffc0000b98,
    bin_shard_sizes=0xffffc0000b08)
    at /usr/obj/usr/src/arm64.aarch64/lib/libc/jemalloc_jemalloc.c:1449
#12 malloc_init_hard_a0_locked ()
--Type <RET> for more, q to quit, c to continue without paging--
```