

# Assignment10

## Team18

112062519 廖思愷

112062636 游竣量

111065547 游述宇

1. Explain how this version of `getenv` achieve thread-safe according to the code above?

- 互斥鎖：使用互斥鎖 `env_mutex` 確保一次只有一個 `thread` 可以訪問環境變數 `environ`。這樣可以防止發生 `Race Condition`，例如兩個 `thread` 同時讀取或修改環境變數。
- 線程特定（私有）數據：使用 `pthread_key_create()` 函數來創建線程特定數據 `key`。使用 `pthread_getspecific()` 和 `pthread_setspecific()` 函數來操作 `key` 存儲和檢索每個 `thread` 獨有的緩衝區 `envbuf`。確保每個 `thread` 都有自己的環境字符串副本，防止多個 `thread` 嘗試使用相同緩衝區時可能發生的數據破壞。
- 一次性初始化：使用 `pthread_once()` 函數和 `init_done` 來確保線程特定數據只初始化一次，無論有多少 `thread` 調用 `getenv()` 函數。
- 內存分配：為每個 `thread` 動態分配一個緩衝區，確保每個 `thread` 都有自己獨立的緩衝區。

2. Is it possible to make this `getenv` function async-signal safety by just temporarily blocking signals at the beginning of the function and then restoring the previous signal mask before the function returns? Explain.

要滿足異步信號安全（async-signal safety）有以下兩個條件：

- 如果一個函數是異步信號安全的，意味著它可以在信號處理函數中被調用，而不會引起程序狀態的不一致或導致數據破壞。這種函數必須避免使用全局狀態、進行動態內存分配、或調用其他非異步信號安全的函數等操作。
- 如果一個函數是異步信號安全的，它必須滿足在接收到信號並觸發信號處理函數時，能夠安全地被中斷和繼續執行的特性。異步信號安全的函數在被信號中斷時不會導致程序的不穩定或數據的不一致。

綜合以上，異步信號安全的函數既要能在信號處理函數中安全執行，也要能在接收到信號時安全地處理中斷和恢復。

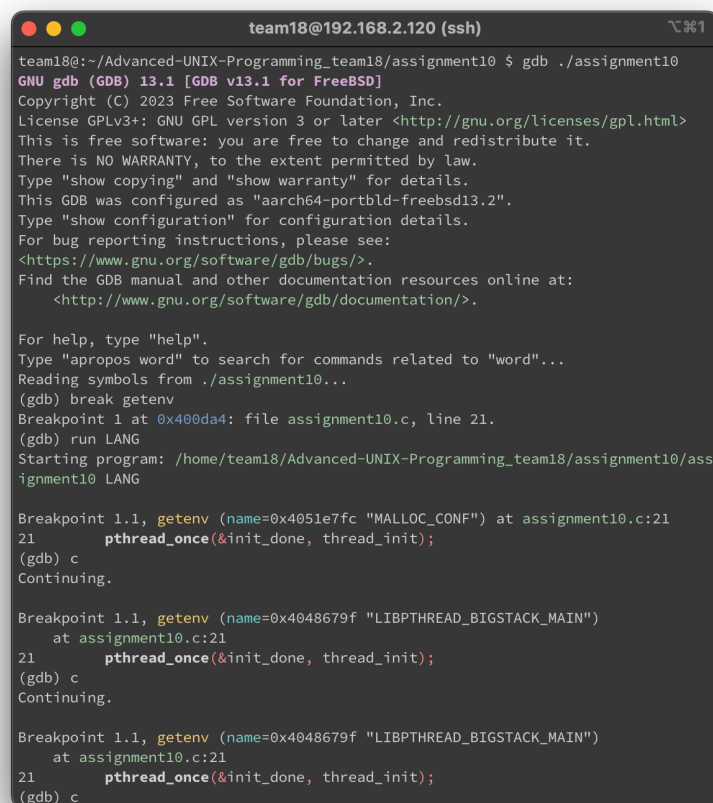
暫時阻塞信號是一種控制技巧，用於保護特定的代碼區塊，以防止特定操作在不恰當的時機被中斷。然而**僅通過在函數開始時阻塞信號，然後在函數結束前恢復信號掩碼，並不能保證使 `getenv()` 函數異步信號安全**。因為異步信號安全要求函數在執行過程中不能使用任何可能改變程序狀態的操作，例如互斥鎖（可能導致死鎖）、靜態或全局變數（可能導致數據競爭）和動態內存分配（可能導致記憶體管理問題）等。而 `getenv()` 函數在執行過程中可能會訪問和修改全局狀態（例如環境變數），在信號處理函數中進行這樣的操作是危險的，因為它可能會與主程序的其他部分（可能正在使用相同的環境變數）產生衝突，從而導致程序狀態的不一致或數據破壞。

3. Please run assignment10.c on FreeBSD and see if it can run successfully. If not, try to explain what happened. You can use gdb to help you find out why and where it crashes.

在 Mac m1 上可以正常執行 `./assignment10 LANG`，而在 FreeBSD 上會陷入無窮迴圈。原因是因為程式碼中重寫了 `getenv` 函數，而在 C 中，`getenv` 是標準庫函數，用於獲取環境變量的值，因此同名的兩個函數發生了衝突。

在此情形中，編譯器可能會使用自訂的 `getenv` 版本而非標準庫函數版本：

- 函數呼叫衝突：由於程式碼中覆寫了標準函式庫的 `getenv`，程式中任何期望呼叫標準 `getenv` 的地方都會呼叫自訂的 `getenv` 版本，這可能導致不符合預期的行為。
- 遞歸呼叫：由於自訂 `getenv` 函數在實作中直接或間接呼叫了標準函式庫的 `getenv` 或依賴標準函式庫的其他函數，這些函數又呼叫了 `getenv`，而導致無限遞歸，最終引發 `stack overflow`。



```
team18@192.168.2.120 (ssh)
team18@~/Advanced-UNIX-Programming_team18/assignment10 $ gdb ./assignment10
GNU gdb (GDB) 13.1 [GDB v13.1 for FreeBSD]
Copyright (C) 2023 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 "aarch64-portbld-freebsd13.2".
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 ./assignment10...
(gdb) break getenv
Breakpoint 1 at 0x400da4: file assignment10.c, line 21.
(gdb) run LANG
Starting program: /home/team18/Advanced-UNIX-Programming_team18/assignment10/assignment10 LANG

Breakpoint 1.1, getenv (name=0x4051e7fc "MALLOC_CONF") at assignment10.c:21
21      pthread_once(&init_done, thread_init);
(gdb) c
Continuing.

Breakpoint 1.1, getenv (name=0x4048679f "LIBPTHREAD_BIGSTACK_MAIN")
at assignment10.c:21
21      pthread_once(&init_done, thread_init);
(gdb) c
Continuing.

Breakpoint 1.1, getenv (name=0x4048679f "LIBPTHREAD_BIGSTACK_MAIN")
at assignment10.c:21
21      pthread_once(&init_done, thread_init);
(gdb) c
```

On Mac:

```
~/Desktop/碩一上/UNIX/Advanced-UNIX-Programming_team18/assignment10 main ./assignment10 LANG
LANG = en_US.UTF-8
```

On FreeBSD:

```
team18@:~/Advanced-UNIX-Programming_team18/assignment10 $ ls
Makefile      assignment10.c      assignment10.pdf
team18@:~/Advanced-UNIX-Programming_team18/assignment10 $ make
gcc -g -pthread -O2 -pipe -c assignment10.c -o assignment10.o
gcc -g -pthread -std=c11 -O0 -Wall -o assignment10 assignment10.o
team18@:~/Advanced-UNIX-Programming_team18/assignment10 $ ./assignment10 LANG
^C
^C
^Z
^Z
[1] + Suspended                  ./assignment10 LANG
team18@:~/Advanced-UNIX-Programming_team18/assignment10 $
team18@:~/Advanced-UNIX-Programming_team18/assignment10 $ ps
  PID TT  STAT      TIME COMMAND
  1227 v0  I+    0:02.06 -sh (sh)
 94925 0   Ss    0:00.36 -sh (sh)
 95475 0   I     0:00.01 /usr/local/libexec/git-core/git credential-cache--daemon
 95556 0   T     0:39.65 ./assignment10 LANG
 95559 0   R+    0:00.00 ps
team18@:~/Advanced-UNIX-Programming_team18/assignment10 $ kill -9 95556
team18@:~/Advanced-UNIX-Programming_team18/assignment10 $ ls
Makefile      assignment10.c      assignment10.o
assignment10  assignment10.core   assignment10.pdf
[1] Segmentation fault (core dumped) ./assignment10 LANG
```

解法：將 `getenv` 函數名改成 `agetenv` 則可正常執行。

```
team18@:~/Advanced-UNIX-Programming_team18/assignment10 $ ./assignment10 LANG
LANG = C.UTF-8
```