

Stk32 C 層當機(閃退)捕捉器 — crash log + MiniDump + 自動上傳

專案:HBSTK 日期:2026-06-10 檔案:c_code/crashcatch_c.c,
build/stk32.hbp, source/main.prg

一、訴求與現象

Harbour 的 ErrorBlock/DefError 只能攔「Harbour 執行期錯誤」(陣列越界、型別、檔案/網路等),這類通常會跳錯誤對話框。但 C/OS 層的存取違規(access violation/GPF)、堆疊溢位會讓 process 在 Harbour handler 跑之前就被 OS 殺掉——畫面瞬間消失、毫無 log,這才是典型「閃退」,原本完全抓不到。需求:讓真正的閃退也能留下證據。

二、解決方法

元件	作法
例外過濾器	Win32 SetUnhandledExceptionFilter 安裝 top-level 過濾器,當機(process 死亡前)被 OS 呼叫。
crash log	寫 _crash.txt:時間、EXE、電腦/使用者、KMSN/站別、例外碼+名稱、出事位址、所屬模組+base+offset、thread、完整暫存器快照。
MiniDump	LoadLibrary("dbghelp.dll")+GetProcAddress("MiniDumpWriteDump") 動態載入,寫 .dmp(免額外連結 lib),事後可用除錯器看完整堆疊。
自動上傳	寫 .trigger(TYPE=CRASH, FILES=...)到 error_pending\,沿用既有 StkAgent 機制上傳到 /H/stk-error/{KMSN}/{STATION}/,再經 ErrorMonitor 通知 Telegram。
安靜結束	回傳 EXCEPTION_EXECUTE_HANDLER + SetErrorMode, process 安靜終止,不顯示 WER 當機視窗。

三、踩坑與陷阱

陷阱	內容
handler 內不可碰 VM	當機當下 heap 可能已壞, handler 只用 Win32 API + 固定堆疊緩衝, 不呼叫 Harbour、不配大記憶體; 環境資訊(工作目錄/電腦名/使用者)於啟動時先快取。
BCC -01 假錯誤	測試 hbp 用預設 -01 觸發 BCC 把 SDK 標頭 winnt.h/winioclt.h 的 W8019/W8004 warning 升成 error(假的「1 errors」); 加 -optim- 即過。production stk32.hbp 本就 -optim- 故一次就過。
.c 也禁 Edit/Write	本專案 settings deny Edit/Write 改 .c/.h/.PRG; 新 C 檔以 Python bytes 寫入, 中文註解 .encode("big5")。
KMSN/站別取得	C 層不便讀 Harbour 變數; trigger 的 KMSN/STATION 留空時 StkAgent 會以自身設定補上(已驗證 fallback), 故路由仍正確。

四、使用技術與用法

技術	用法
安裝	main.prg Main() 最前 STK_INSTALLCRASHCATCHER(); 載 INI 後 STK_SETCRASHINFO(GETC("KMSN"), "")。
驗證方法	獨立 test exe: STK_INSTALLCRASHCATCHER() 後以 NULL 寫入觸發 0xC0000005, 檢查 error_pending\ 產生三檔。
驗證結果	攔到 ACCESS_VIOLATION, crash.txt 暫存器 EAX=00000000(NULL 指標)、trigger FILES= 正確、dmp 檔頭為 MDMP, process 安靜結束不跳 WER。
事後分析	.dmp 可用 WinDbg / Visual Studio 開啟, 配合 .map/.pdb 還原完整呼叫堆疊與崩潰行。

結論: 原本「真正閃退抓不到」的缺口已補上, 與 Harbour 層 DefError 互補, 覆蓋兩類當機。