

著作權潔淨室

COPYRIGHT CLEAN ROOM

陳 歆 撰

背景

潔淨室在台灣誕生肇因於1988年 IBM 向宏碁電腦索取九百萬美元著作權侵害賠償。當時個人電腦產業景況可用「霸權」來形容；藍色巨人生產的機器奠定了個人電腦技術規格的標準，因而 IBM 獨佔了電腦的基礎輸入/出介面系統(即所謂 Basic Input/Output System, BIOS)，任何擬進軍 IBM-Compatible 機器的公司，必定要跟隨 IBM 的 BIOS 系統。

IBM 控告宏碁侵害其 BIOS 中的鍵盤控制器(keyboard controller)的密碼著作權，初步要求賠償金額高達三千萬美元。在退出 IBM 相容電腦或支付天文數值賠償的兩難選擇，宏碁只好咬緊牙關付給 IBM 協議金額的九百萬美元。

事後，宏碁的成本/效應分析結果自然選擇「預防勝過救急」的措施，這就是建立潔淨室的緣故。潔淨室的建立，原為迴避官司的消極措施，然而因同時給予軟體研發部門無後顧之憂的研發自由，亦顯現其積極作用；

新的電腦程式規格一開定，若無畫面「整體觀感」的實質相似問題，國內有才能的程式設計師可撰寫自家的新產品。如此，著作權潔淨室非但為防禦的甲盾，亦是進攻新商機之利器。

知道潔淨室的作用，卻不使用，應算是經營者策略上的一大過失。美國第二大微處理器廠商 AMD 於 1992 年與英特爾官司敗訴後，業者認為其主要錯誤是沒有及時成立撰寫 486 微密碼的潔淨室，導致股票在兩天之內有 45% 之跌幅。

潔淨室的法律作用—舉證責任的轉移

潔淨室程序所產生的種種指令及記錄文件在法律上確可反駁侵害著作權的指控，單是其系統的存在就會轉移法律上所謂的「舉證責任」。一般的著作權侵害指控成立之後，舉證責任就在被指控的一方；換句話說，被指控者要證明他沒有抄襲某程式或密碼。然而，就著作權侵害的認定基準而言，即使被指控者確實無罪，證明

其清白並非容易；被指控者若有建立成的潔淨室系統，則舉證的責任就轉移到對方。指控者需要證明這潔淨室執行的不當。因為相關潔淨室的資訊、文件、資料等均在被指控一方的控制之下，指控者很難證明潔淨室的種種不是。更且，因潔淨室的存在足以顯示對著作權侵害法律的在行，對手通常就會知難而退。國外的智權律師很清楚法律的訴訟程序以及舉證責任的重要性的舉證的困難；因而面對一個潔淨室系統，很多侵害案就會在起跑前棄權。此乃「形式重於實質」之另一實例，純是潔淨室存在所產生之實質結果。

潔淨室程序會產生有系統的技術和法律資料，有時會有想不到的作用。宏碁曾被告剽竊營業祕密，潔淨室的資料就當了先發明的證據。因為潔淨室的資料完全系統化，在法律舉證作用上頗具說服力。

潔淨室的技術開發作用——開發軟體的系統化

潔淨室的記錄文件另有一個非常重要的作用：就是新軟體開發工作進度的系統化。潔淨室程序的記錄有以下的重要作用：(1)讓主管及程式設計師回顧project (包含記在會議記錄的策略及更改過程等)的各種進展情況；(2)

因新人可參考開發記錄就能迅速地瞭解project 以及自己的工作任務，故讓參加project 的新人很快就進入情況；(3)幫助工作交接；(4)幫助project 的責任分界。

潔淨室的法律基礎——接觸機會

著作權侵害是基於電腦程式的「實質相似」(substantial similarity)和程式設計師是否與原作原始碼有「接觸機會」(access)。相容性軟體，因功能及工業標準的需求，相似度必然很高，故難避開實質相似的判斷；故侵害與否的關鍵在於接觸機會的判斷。簡單地說，潔淨室的作用就是來否決接觸機會，若能表明無接觸機會，就無抄襲的可能，乃無著作權侵害。

潔淨室的組織

潔淨室執行人員有分規格(Specification)、撰寫原始碼(Coding)、及法務(Legal)三組。因撰寫程式碼的工程師(撰碼組，Coding Team)必定與涵蓋軟體表達方式之所有相關資訊徹底隔絕，規格組(Spec Team)所提供的規格必須取得法務組(Legal Team)的核准，才能過關傳達給撰碼組。此隔絕和核准的程序，固然會耽擱時間和帶給規格組及撰碼組若干麻煩，可是若

相較於該軟體產品，因著作權問題而上不了市，或公司被告著作權侵害，導致整個 project 所投資的時間及金錢全部報廢，潔淨室程序在開發新軟體所增加的若干措施，終究應該是值得。

法務組必知

法務組如何判斷哪一種資訊/資料可過關？且須瞭解若干著作權法的基本原理。首先，過關或攔住的最基本考量乃為著作權最基本的原理：構想與表達方式的分歧 (idea/expression dichotomy)。簡單地說，構想資料可通關，表達方式資料必被攔住。說來單純，不過執行分辨工作經常屬形而上學；美國知名法官 Learned Hand 曾說過：「從沒有人指定過此歧野，而沒有人能指定得到(Nobody has ever been able to fix that boundary, and nobody ever will)。」電腦程式的原始碼(source code)就是構想的表達方式(Data Cash v Personal Micro)，決不得通過；程式的規格(spec)可通過，但不得太「緊」(tight)。何謂「緊」，何謂「鬆」？大致而言，撰碼組，閱讀該規格之後，若大概能斷定逐一行的原始碼如何寫，該規格就太靠近程式的表達方式，故為太緊。

構想／表達方式的分野之外，尚有執行潔淨室須知的相關法律推論：

例如，構想與表達合一(Merger)、表達不可或缺的佈景 (Scenes a Faire)、表達選擇有限 (Limited Choices)、實用功能性 (Functional Works)、及空格表(Blank Forms)等著作，均不得享有著作權保護。

合一論經常出現在法官的判決書；一個較明顯的案例是七十年代的大型電腦使用的電腦程式打卡 (computer punch cards)；打卡的構想及其表達方式(就是表達電腦程式帶有洞的卡片)無法分開，因而此電腦打卡著作權登記申請案以「合一論」為理被駁回。Synercom v University Computing.

「不可或缺的佈景」(Scenes a Faire, 法文)，係指構想所不能或缺的情節、角色、或環境等佈景。因不能或缺，其構想和表達方式就無法分開（類似合一論），故不得享有著作權保護。頗受歡迎的早期電腦遊戲「小精靈」(Pac-Man)的構想所不能缺少的遊戲迷宮 (maze)，為「小精靈」的不可或缺的佈景。Atari v North American Philips.

實用性作品 (Functional Works)，本為專利的領域，人文藝術品為傳統著作權的保護領域，故原則上，實用性的作品就不得享有著作權保護。蓮花(Lotus)軟體開發公司的工作試算表

(spreadsheet) 和主功能表 (menu) 上的指令(menu commands)在初審被判係著作權保護範圍內(*Lotus v Borland*)。可是,「蓮」花一現,功能表的指令遭上訴法院(聯邦上訴法院,CAFC)改判為「實用性質」,故不得著作權保護。經1997年再上訴,美國最高法院不予受理,因而CAFC的判決就正式成立。因為「實用性質」的認定極不清楚,利用此著作權的例外較為困難;一個電腦程式本身就是「實用性」的,但其整體表達就會受著作權的保護,因此大概只能在程式中形容某功能為「純實用」為辯論的基礎而引用*Lotus v Borland* 案例。

極有限的表達方式選擇之作品不得享有著作權。一個地圖為例:表達街頭的相對佈置沒有選擇的餘地,若此唯一的表達方式給予著作權,乃必會妨礙地圖藝術/技術的發展,因而純街頭佈置的表達無著作權保護。不過,地圖若附帶一些特特殊設計,例如美化的字體,或車站、旅館、加油站的資訊,則此加以表達的地圖可升為受著作權保護的作品。另一例子是電話簿裡的名單排列;純照字母順序或筆畫排列,並無選擇的餘地,可是若加一些小廣告或行業分組等特徵,則有升為著作的可能(*Asian Yellow Pages*)。最近,電腦資料庫 (databases) 的著作

權保護問題引起著作權法界的激烈爭辯,此問題亦為選擇有限的著作權課題。

帶空格的表格不受著作權保護,有早年的權威案 (*Baker v Selden*, 1880年,會計帳本表格) 為最早的案例。以上所述的試算表,若無主控指令,肯定不受著作權保護。但是,任何表格若有資訊傳達的作用,而其資訊傳達帶有原創表達性質,可能會升為受保護的著作。

法務組的執行方法

法務組人員應基於以上所述的法律原理對某資訊/資料作「過關/攔住」的判斷。其判斷程序簡單地說可照以下的步驟:(a)構想與表達方式能不能分開(合一論, Merger)、(b)表達方式是否係不可或缺的佈景 (*scenes a faire*)、(c)表達方式是否係選擇有限 (limited choices) (此包含工業標準 (industry standard)的表達規格)、(d)功能是否屬實用性 (functional)、(e)資訊/資料是否僅提供空格表(blank forms)的資訊而已、(f)資訊/資料是否已被認定為公用資訊(public domain documents)。若擬進入潔淨室的資訊/資料能歸類於任何(a)至(f)六項,因不受著作權保護,可安全過關。此判斷程序就如同目前法律最具權威性的「抽象、篩選、比

較」侵害判決基準 (abstraction-filtration-comparisons test)。此判決方式含有四個步驟：(1)解剖程式組合，分成個別部分；(2)濾除各部分的含有構想 (constituent ideas abstraction)、構想所不能或缺的表達方式 (*scenes a faire*)、構想與表達方式之合一部分 (merger)、和已成為公用部分 (因已被貢獻為公用而權利消滅，public domain elements)；(3)篩選區分剩下的部分 (filtration) (此為受保護的表達方式部分)；(4)把篩分出來的部分與被告的程式相比 (comparison)；如係實質相似，若有接觸機會，就有侵害著作權之嫌 (*Computer Associates v. Altai*)。值得一提的是，*Altai* 一案似乎更進一步地正式加以肯定：利用潔淨室程序重寫曾被污染的程式能挽救抄襲的侵害罪名；換句話說，只要遵守潔淨室程序，有罪者有救，浪子可回頭。

另外，有時供應商會願意免費提供可直接抄襲的程式原始碼。此尤其會發生在其推廣自創的工業標準或在推銷新產品活動的情況。若擬寫信問清楚，必須謹慎，若被拒絕，將來撰寫類似的程式就可能已經有打草驚蛇的風險。當然，若無其他法，可要求受權而支付權利金。

篩選程式中受著作權保護部分之後，法務組另外必須注意兩項極重要

的法律推論：即衍生作品 (Derivative Works) 及合理使用 (Fair Use Doctrine)。

英美法的衍生著作理念原是把原作者的著作權延展到所有基於原作的作品；也就是說，沒有獲得原作者的同意，他人不得基於原作而著作。有趣的是，國內著作權法的「衍生著作」定義正好相反，「衍生著作」為「原著作改作之創作，以獨立之著作保護之」（*中華民國著作權法第六條*）。換句話說，英美法的衍生著作權屬於原作者，國內法的衍生著作權屬於改作的作者。這差異或許能反映先進國家較嚴格保護原作者的意圖，而開發中國家偏於較自由利用原作權利的趨向；也就是說，先進國家擬保護其開發的較為基礎性之創造程式，而開發中國家擬將保護重點放在其比較會研發的有附加價值之改作創造。但無論係西方或東方的觀念，其共同點是衍生著作若有加以實質的變更 (substantial variation, 一種附加價值)，該衍生著作的著作權屬衍生作者自己。如此，國內廠商在研發基於他人程式的軟體過程，為避免著作權侵害之嫌疑，必須儘所能地改進（最好是實質改良）原作程式，即有自己的著作權。不過，此權利可能仍然基於原作；原作若不係以上的不受著作權保護的作品，一

般需要原作者的授權才能銷售。軟體新版本和修改過的軟體(modified)為最普遍的衍生著作。因衍生著作的作者須原作者的使用同意，原作者若對衍生著作的改良特徵(new features)有興趣，兩全其美的協議是建立軟體使用的相互授權關係。法務組和程式設計師可合作判斷衍生著作是否帶有實質的變更；若是，含有自家的衍生著作應無侵害著作權之虞。

合理使用的推論原為紓解一些著作權法執行方面的困難(例如，如何控制原創作品的影印重製)和避免妨礙教材資料的普及化。合理使用論的初步考量是：一般以教育為目標的重製係合理實用，因而並非侵害著作權行為；反而以商業（營利）為目標的重製或使用一般視為著作權侵害。當然教育與商業行為的分野有時模糊；比方一個公司的程式設計師所重製的「程式設計新知」文章發給同事，然後從該文章所學的「新知」被應用在公司的新產品中；此重製行為係教育或商業性質？恐怕只有看律師的辯論能力才能解決。合理使用有另一重要的應用地方：還原工程。

還原工程(Reverse Engineering)

電腦程式的「還原工程」包含「反編譯」(decompilation)及「反組合」

(disassembly)；前者為目的碼(object code)的所反編譯(decompile)成原始碼(source code)；後者為目的碼的所反組合(dissassemble)成電腦合成語言(assembly language)。反編譯是將目的碼本身的一系列「0」與「1」編譯成程式的種種概念，結構、順序、組織等。雖然目的碼的「0」、「1」型態無法辨認出任何「表達」，其照樣受著作權之保護(*Tandy v Personal Micro*)。此判決曾引起了一個基本觀念的問題。*Personal Micro*的抗辯書指出：目的碼乃為電腦程式中的每一行指令編譯後的「0」與「1」代之，著作權保護的是構想的表達方式；可是沒人能看得懂此表達方式，何謂expression？怎能給予看不懂的著作著作權？對此抗辯，法官靈感地反駁，人類可能看不懂目的碼，可是電腦機器不但能看得懂，也會聽命行事，因此可視為一種表達方式，因而完全符合著作權法理。

依照權威案例，原始碼(source code, *Data Cash v Personal Micro*)、目的碼(object code, *Tandy v Personal Micro*)、和微密碼(microcode)均受著作權保護。雖然所謂韌體(firmware)似乎在專利與著作權之間，晶片中所鍍寫的microcode在*Apple v Franklin*一案被判為著作權保護範圍之內。但，往

後的案例指出：潔淨室的運用，在撰寫晶片裡的微密碼，可避開著作權侵害之虞。

反編譯的過程必須以潔淨室程序進行之。複雜性質頗高程式的反編譯工程需付出相當的工夫和專業知識才能完成。除了完全屬實用性或已經成為工業標準的部分，因而不受著作權保護之外，反編譯工程在依據目的碼所擬定的新原始碼時，需要在繁雜的程式設計種種途徑中，主觀性地選擇程式步驟；是以，還原工程所產生的新程式必定會有其獨特的表達方式。因而，反編譯工程似乎符合著作權法的基本原理及推論：程式的構想 (idea)、實用功能 (function)、工業標準 (industry standards) 以及其餘 (public domain、merged、scenes a faire、forms) 不受著作權保護，可以抄襲。通常一個程式不受保護部分佔了密碼的一大半，其餘的密碼不得抄襲，而可供程式設計師表達才能的機會；此新創造（或衍生）著作就是新產品的結晶，應享有自己的著作權。

以重製電腦程式為索取其中的概念為目標的還原工程被判為一種合理使用 (*Sega v. Accolade*)，因而並非著作權侵害；但是，還原工程以抄襲密碼為破開封鎖系統的商業利益為目標被判為侵害著作權的非合理使用行為

(*Atari v. Nintendo*)。不過，判決書指出，還原工程的目的和性質均應考慮為判決因素。合理使用的判斷大致可分為商業性質(非合理使用)和教育性質(合理使用)。電腦軟體的重製和銷售當然屬商業性質；可是，照此案的判決理念，如果反編譯工程的結果會導致更多有創意價值的相容性產品，這種商業性行為可能會被視為合理使用；此解釋雖然模糊了合理使用的理念，其用意當然不外於是鼓勵技術改良的努力。

整體觀感(Look & Feel)的陷阱

儘管潔淨室的嚴格執行，若模擬的程式為一市場上的流行應用程式 (application programs, AP)，而該 AP 具有獨家的特殊螢幕畫面設計，若模擬程式的「整體觀感」(look & feel)與原作實質相似，該模擬程式恐怕仍有侵權的嫌疑。舉例，新開發的程式畫面若與 Windows 95 的畫面相似，因 Windows 95 的普遍使用，雖然執行過嚴格的潔淨室程序，接觸機會仍會被推定(presumed)；因而，潔淨室，在開發帶有特殊花面設計的流行應用軟體程式，無用武之地。碰到如此的法律堡壘，國內軟體開發公司，在美國全然占領應用程式市場之情況，很難找出一席之地；恐怕只有在創造全新的

程式求發展，例如中文或東方文化為主的特殊應用程式。

潔淨室的執行

以上所述的理論係為建立執行的基礎。執行措施煩瑣，必須有組織、有條有理地進行；流程進度一失去紀律，曾辛苦建立的制度有一旦全泡湯之危。

因撰碼組的角色比較敏感，為避免選撰背景污染的程式設計師（background contamination），規定所有新進公司的撰碼組人選必須到台北地方法院公證宣誓未曾接觸過類似要撰寫的程式，並須切結「絕不抄襲第三者相關原始碼及目的碼，亦不會仿冒或刻意侵害他人的著作權」。新人選亦必須在切結書上附加「所有電腦工作經驗皆摘要」資料。為了加強新人選的法律觀念，法務組應定時舉辦「著作權法」一課，規定撰碼組新成員必須參加聽課。

為防備污染（contamination prevention），潔淨室必須具備有系統性的記錄文件；例如：進出潔淨室的簽名及時間記錄、「每日工作記錄」（daily log）、程式更改申請（engineering change requests, ECR）、程式更改及修改記錄（revisions and replacements record）、程式瑕疵清單（bug list）、工程筆記本

（engineering notebook）、會議記錄（meeting minutes）、仲裁會議記錄等。這些記錄文件就構成文件途徑資料（即所謂 paper trail），其作用在於證明獨立開發。記錄文件亦係極有效的具體證明資料（evidence of a system and procedures for protecting copyright）。

宏碁研發的「潔淨室程序機器」（CRP MACHINE）有提供潔淨室執行的流程以及所需要的標準文件和表格。因為法務組「緊」與「鬆」的判斷是執行之關鍵，而最為困難的一步，宏碁另外亦有基於十年執行經驗的著作權法基本原理及推論為主的「潔淨室法務組交互操作判斷程式」（Clean Room Legal Team Interactive Decision-Making Program）；其功能可使法務組的「過關」及「攔住」的判斷幾乎自動化。

比對程式

為了更進一步證明新開發完成的軟體將來無著作權的問題，應否決著作權侵害的另一個基石：「實質相似」（substantial similarity）。新程式若是獨立開發（或密碼多半係自家撰寫），宏碁研發成的「電腦軟體比對程式」可用來證明。八十年代，美國海關所用的類似「Gumshoe」及「Side by Side」比對程式，因軟體的發展，現在無大

用。過去的 compilers 種類無多，亦無多 optimization 或 compression 的處理程式。如今的比對程式必須考慮到新技術的發展才能精確的完成有價值的比對。簡單的說，一個比對程式會逐步比對每一行的目的密碼，處理「添補」(filler)以及工業標準部分之後，就會計算相似比例的百分值。依據著作權法，極低的百分比無確定不侵害的作用；但是，比值仍可當初步的侵害答辯反駁，至少能把舉證責任推至對方。值得一提的是，因較低的相似度有利於侵害反駁，法務組應勸告撰寫程式的人員，有機會就務必多使用自家(或個人)的填補密碼(filler)，千萬不得借用他人的 filler 密碼。

最後，雖然伯恩公約（著作權的國際組織）規定毋需註冊著作權（作品一完成就享有著作權）新完成的軟體仍應盡快去註冊。如此至少可初步奠定所有權及獨立開發的假定；萬一被告，註冊證明亦可當辯護的證據。若發生他人有抄襲自家程式之嫌，著作權侵害的基本起訴文件之一乃為著作權所有權證明；註冊書就是法律所需的基本資料。另外，註冊證明可有效的反駁「不知情」(innocent infringement)的侵害辯護（註冊視為公開文件，相關人被假定有所知）。當然國內目前對著作權不受理，開發新軟體公司可考慮在其他國家註冊（如美國，註冊規費不高，手續簡單，約三個月就下來了）。

(作者任職宏碁電腦股份有限公司
智慧財產組法務長)