

Unix: Història i filosofia

Història

Eren els 60 i els laboratoris Bell d'AT&T junt amb el MIT (Massachusetts Institute of Technology) treballaven en un sistema operatiu experimental anomenat Multics (Multiplexed Information and Computing Service), dissenyat per funcionar en un GE-645, un potent ordinador d'aquella època. Les proves de rendiment de les primeres versions d'aquest sistema no van anar tan bé com s'esperava per la qual cosa els laboratoris Bell van paraitzar el desenvolupament. Ken Thompson i Dennis Ritchie, que eren els principals responsables del projecte, es van quedar sense res que fer i morts de l'avorrimt van decidir crear un joc, el Space Travel, que funcionava sobre aquell GE-645. La joguina els agradava, però t'nia l'inconvenient de què sol es podia jugar en aquell "supercomputador" tan car així que van decidir portar-lo a una màquina més petita, una PDP-7. Això no era tasca fàcil ja que en aquell temps tot es desenvolupava en llenguatge ensamblador i per aconseguir el seu objectiu calia fer canvis en la gestió de memòria, afegir suport per a nombres reals, ajustar-se al sistema de fitxers, etc. Tant es van embolicar en aquesta tasca que van acabar creant un nou sistema operatiu, l'UNICS. AT&T, que va veure que la cosa podia tenir futur va decidir subvencionar el desenvolupament incorporant a diversos programadors, entre ells a Brian Kernighan. UNICS passa a dir-se UNIX i va anar adquirint noves funcionalitats i petits programes que s'incorporaren al sistema com a ordres. El 1971 ja es podien usar, per exemple, cat, chmod, chown, cp, ls, mv, wc, who, roff...

El desenvolupament era ràpid però hi havia una cosa que el frenava. Cansats de programar en ensamblador, Kernighan i Ritchie van decidir crear un nou llenguatge de programació de nivell més alt, afegint noves funcionalitats al ja existent llenguatge B. El resultat va ser C, un llenguatge, que permetia abstrèure's de les particularitats de l'arquitectura subjacent però al mateix temps treballar a nivell de bit i gestionar els recursos de memòria. La seva potència i elegància van impressionar tot l'equip de desenvolupament per la qual cosa al 1973 es decideix reescriure en C tot UNIX. Molt poc després, en 1974, Ritchie i Thompson presentaven a la comunitat el seu sistema a través d'un article publicat a la prestigiosa revista Communications of the ACM, on parlaven sobre el disseny d'Unix i sobre les 600 màquines que de moment funcionaven amb ell. És llavors quan el sistema operatiu comença a guanyar popularitat i el seu desenvolupament s'accelera vertiginosament. Sorgeixen múltiples versions i es porta a una infinitat d'arquitectures diferents. Les universitats comencen a utilitzar-lo com a plataforma de treball per a la investigació, grans empreses com bancs i companyies d'assegurances l'empren com

a base dels seus sistemes i fins i tot els exèrcits comencen a experimentar amb ell per adaptar-lo a les seves necessitats. AT&T introdueix el seu Unix System 3, la universitat de Berkeley desenvolupa el BSD v4 i alguns venedors com Interactive Systems, Microsoft o Human Computer Resources distribueixen versions adaptades a ordinadors més modestos.

Totes eren versions plenament funcionals però mancaven d'un suport de xarxa adequat. Els experiments amb Ethernet no havien sortit del tot bé i l'únic que existia era un servei molt pobre que permetia compartir dades i reexpedir correu: l'UUCP (Unix to Unix Copy Program), que havia estat desenvolupat pels laboratoris Bell per distribuir programari a través de mòdems connectats a la línia telefònica convencional.

El departament de defensa nord-americana acabava de finançar el desenvolupament de la pila de protocols TCP/IP i necessitava un equip per implementar-la a la seva xarxa experimental ARPANET. L'equip de la universitat de Berkeley estava en el moment adequat, al lloc adequat i va ser elegit per dur a terme aquesta tasca. Aquest fet és probablement el punt d'inflexió més gran que ha experimentat Unix en tota la seva història, ja que comptar amb una implementació de TCP/IP dotava al sistema operatiu de la funcionalitat que li faltava i que tant èxit li ha suposat: la connectivitat. Amb els *sockets* de Berkeley, Unix comptava amb una interfície senzilla que permetia enviar de forma fiable informació des d'un ordinador a un altre, fins i tot quan ambdós estaven connectats a diferents xarxes. Això suposa el boom definitiu d'Unix, que s'alça llavors com el sistema operatiu omnipresent en tots servidors de xarxa i supercomputadors del món.

Ja en la dècada dels 80 són dos els fets que marquen negativament la trajectòria d'Unix.

Per una part, el 1981, Bill Gates compra a Seattle Computer Products el QDOS (Quick and Dirty Operating System) per 50.000 dòlars, el que resulta ser una de les millors inversions de la història ja que els beneficis que li reporta la comercialització d'un programari no escrit per ell i rebatejat com MS-DOS són de l'ordre de bilions de dòlars. Microsoft aconsegueix així el control de tot el mercat dels ordinadors personals que, en no comptar amb el maquinari adequat per fer funcionar Unix, havien de conformar-se amb sistemes operatius de categoria molt inferior, com el subministrat per Gates.

D'altra banda, sorgeixen batalles legals entre la universitat de Berkeley i AT&T. En comprar la llicència d'AT&T, la universitat volia experimentar amb aquell sistema operatiu modern que incloïa codi font. L'equip d'investigació escrivia desenes d'utilitats noves i les redistribuïa amb el nom de BSD (Berkeley Software

Distribution). En principi tots els canvis eren distribuïts a posseïdors d'una llicència d'AT&T però arriba un moment en què la universitat canvia tants aspectes interns del sistema operatiu que la diferència entre les dues versions es fa enorme i Berkeley considera que pot redistribuir-lo com a propi a qualsevol persona que no posseeixi la llicència original. Això enfurisma AT&T que demanda Berkeley per infracció de copyright, que al seu torn demanda AT&T per copiar fragments de codi. Comença llavors una enorme batalla judicial en la qual ambdues parts s'acusen mútuament i que es prolonga fins a 1994 quan Novell, que dos anys abans havia adquirit tots els drets de l'Unix d'AT&T, firma la pau amb la universitat de Califòrnia. L'acord adoptat consistia que Berkeley havia d'eliminar tot el codi residual d'AT&T i llançar una última distribució de BSD totalment lliure de problemes de llicenciament. Aquesta distribució va ser la 4.4-BSD Lite2.

En meitat de tota aquesta batalla legal un estudiant finlandès anomenat Linus Torvalds, enfadat pels alts preus que tenien els Unix per a x86, decideix crear el seu propi sistema operatiu inspirant-se en Minix, un sistema creat pel professor Andrew Tanenbaum com a recurs pedagògic.

Quan aconseguix un prototip funcional publica un missatge a Usenet:

Sospites pels fabulosos dies de Minix-1.1, quan els homes eren homes i escrivien els seus propis controladors de dispositiu? Ets sense cap projecte decent i et mors per clavar-li les dents a un sistema operatiu que puguis modificar per a les teves necessitats? Et sembla frustrant que tot a Minix funcioni? Se't van acabar les nits tractant d'aconseguir que un programa funcionés? Llavors aquest post pot ser per a tu ;-)

Com vaig dir fa un mes, estic treballant en una versió lliure d'una espècie de Minix per a ordinadors AT-386. Per fi ha arribat a una fase en la que es pot usar (o, no depenent del que vulguis) i m'agradaria publicar el codi font perquè s'estengui. Només és la versió 0.02 però he aconseguit executar en ell bash/gcc/gnu-make/gnu-sed/compress etc.
[...]

El missatge va suscitar un enorme interès en un sector de la comunitat internauta i van ser molts els programadors que van decidir unir-se a aquest projecte, que va adoptar el nom de Linux en honor al seu creador. Torvalds només estava interessat en dissenyar el nucli del sistema operatiu, per aconseguir un producte complet es necessitava implementar tot el conjunt d'eines que tradicionalment havien acompanyat als sistemes UNIX. El projecte GNU de Richard Stallman encaixava com a anell al dit ja que aquest havia estat anys desenvolupant un clon lliure d'UNIX i ja comptava amb tot el conjunt d'eines necessàries. Curiosament a Stallman li faltava implementar el kernel del sistema pel que la unió va ser perfecta i així va sorgir GNU/Linux. A partir d'aquell moment GNU/Linux no n'ha fet més que estendre's a una velocitat

vertiginosa, amb milions d'usuaris al món sencer. Aquest fet ha estat probablement el segon més important en la història del sistema UNIX ja que la seva filosofia de codi obert ha aconseguit estendre el sistema operatiu, i en general el programari lliure, fins i tot fins als ordinadors de sobretaula, terreny que sempre havia estat copat al 100% per Microsoft i Apple.

En l'actualitat els diversos "sabors" d'UNIX són el sistema utilitzat en la majoria dels sistemes de càlcul i experimentació científica, de servidors de xarxa, de control de processos industrials i de supercomputadors en general. Gràcies a Linux també posseeix una quota gens menyspreable al mercat dels ordinadors de sobretaula i es preveu un creixement exponencial, podent a curt-mig termini arribar a convertir-se en un seriós competidor als sistemes operatius de la companyia Microsoft.

Filosofia

Unix ha provat el que cap altre sistema operatiu no pot imaginar. Diferents versions han funcionat sobre supercomputadors, PDAs, maquinari de xarxa, telèfons mòbils, vídeoconsoles, etc., però Unix és molt més que un sistema operatiu. Unix es caracteritza per tenir una filosofia i una forma diferent de veure les coses, així com per ser el sistema operatiu dels entusiastes de la informàtica. La complexitat d'ús que li atribueixen els defensors de sistemes inferiors com MS Windows és vista pels usuaris d'Unix com un dels seus majors atractius. La idea de conèixer el funcionament intern del sistema, la possibilitat de modificar les seves parts, d'afegir nova funcionalitat o la seva enorme fiabilitat i absència de penjades, ha fet que molta gent s'identifiqui totalment amb aquest sistema operatiu i es converteixi en un fervent defensor del mateix.

Igual com Internet, Unix té una cultura pròpia, té un estil de programació característic i porta amb si una potent filosofia de disseny. Entendre el món i la comunitat que envolta a UNIX és essencial per poder desenvolupar programari per a ell. Freqüentment s'acusa aquesta comunitat de ser un grup molt tancat i reticent al canvi. Això és bastant cert i sembla que no canviarà en un futur pròxim pel que és convenient familiaritzar-se amb la filosofia que envolta al sistema per poder integrar-se al seu món.

Bàsicament tota la filosofia de la programació a Unix es pot resumir amb la frase "*Do one thing and do it well*" (fer una cosa i fer-la bé), però en general hi ha una sèrie de pautes que s'haurien de tenir en compte:

- Fes que cada programa faci una cosa i la faci bé.
- Per dur a terme una nova tasca escriu un programa nou. No en compliquis un vell afegint una nova funcionalitat.

- Escriu el teu programa tenint en compte que la seva sortida probablement sigui l'entrada d'un altre programa. No omplis *stdout* amb informació innecessària ni utilitzis formats rars.
- Guarda les dades en arxius de text pla. Si necessites seguretat, confia en els permisos.
- Usa noms curts i en minúscula.
- Si no és imprescindible, no demanis res de forma interactiva: fes que l'usuari subministri les dades per línia d'ordres en la crida.
- Fes parts simples connectades mitjançant interfícies netes i ben definides.
- Centra't en les dades. Si has triat les estructures adequades i estan organitzades totes correctament, els algorismes seran evidents.
- Claredat millor que complexitat. La solució més simple és freqüentment la millor: afegeix complexitat només on sigui indispensable.
- Portabilitat millor que eficiència.
- Pensa en paral·lel. Hi ha altres processos al món, fins i tot instàncies del teu mateix programa funcionant alhora.
- Fes un programa gran només quan hagi quedat demostrat que no pot fer-se amb un petit.
- Si el teu programa no té res interessant que dir, que no digui res.
- No existeix una única manera de fer-ho. Cada problema té múltiples solucions.
- Dissenya pensant en el futur, és més a prop del que penses.
- Unix no demana si us plau.

En cas de dubte la regla universal a tenir en compte és sempre la norma KISS: *Keep it simple, stupid!* (mantenlo simple, estúpid!)