

Programfejlesztés

Olvasd többet! Tudom, ez úgy hangzik, mintha egy mozgókönyvtár plakátjáról kölcsönöztem volna, de komolyan gondolom: olvass többet! Annyiféle eszköz áll a rendelkezésünkre programíráshoz, és oly sokféle helyen tehetjük alkotásainkat közzé, hogy az írás lassan veszélyessé válik. Ezáltal pedig az olvasás sem lesz könnyebb.

Az első tanács, amit minden programozónak célszerű megjegyeznie: ne írd meg azt, amit máshonnan fel tudsz használni! Ez rendben is lenne, csak hogy ahhoz, hogy meg tudjuk ítélni, mit lehet újrafelhasználni, értenünk kell az olvasáshoz. A hatékony kód-újrafelhasználás nem csak annyiból áll, hogy például: „SSL-t akarok használni, ezért rákereksek valamilyen +ssl+ könyvtárra.”

Nagyszerű, megtaláltad az OpenSSL-t. Most hogyan tovább? Miként lesz a segítségedre? Mivel nem egy barlang mélyén laktál az elmúlt tíz évben, elolvasol valahol valamilyen leírást a témáról. Milyen más megoldást – esetleg valami kevésbé közismertet – lehetne használni az SSL-en kívül? Ha nem olvasol, sosem fogsz tanulni, és ismét csak mindent magadnak kell megírnod.

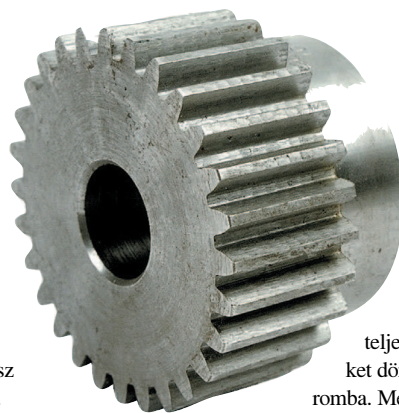
Brian Behlendorf egyszer ezt írta: „Kevesebb programot a világnak!” Ez nem azt jelenti, hogy ne írjunk programokat, amennyiben szükségesek, csupán azt, hogy megfelelő indok hiányában ne írjunk programot. Természetesen eleinte sokkal élvezetesebb lehet saját sorokat írni, mint mások kódját hosszasan böngészni és megérteni, ám egyszer annak a napja is felvirrad, amikor észreveszed, hogy ezt a lenyűgözőnek tartott „új” dolgot már időtlen idők óta írod, és a pusztá gondolatától is rosszul leszel; arról nem is beszélve, hogy újra és újra foglalkoznod kell a karbantartásával. Tanuld meg inkább, hogyan lehet mások munkáját újrafelhasználni, ezáltal arra a tudásra is szert teszel, hogyan dolgozhatnak neked mások. Egy levelezőlista-tag vetette fel egyszer a következő kérdést: „Egy olyan folthoz is szükséges a szerzők nevét feltüntetnem,

amely csupán kódrészleteket távolít el?” A projekt egyébként egy GUI-tervezet volt, a szóban forgó folt pedig néhány alapértelmezett elrendezéssel egy párbeszédablaknál használt tetszetős, ám félkész állapotban hagyott programot módosított. Természetesen ebben az esetben is fel kell tüntetni a szerzők nevét. A folttal ugyanis nemcsak jobbá és gyorsabbá titted a programot, de példát is mutattál másoknak. A kód megtisztítása (a sorok eltávolítása) a kód alapos megértését követeli meg, ami számos esetben egyben még a hozzáadásnál is magasabb fokú hozzáértést jelent – így aki erre képes, annak neve joggal kap helyet a szerzők listájában.

Emlékszel az XCoffeeera? Hatalmas ötlet volt: a név egy olyan különleges célú kiszolgálóalkalmazást takart, amely kizárólag kávéképeket szolgáltatott a Hálózaton keresztül, valamint egy hasonló ügyfélprogramot, amely csak kávéképeket jelenített meg. Egészen leleményesnek számított a maga idejében – de csak kávéképeket szolgáltatni, elegendő az? És csak egyetlen képet adni ügyfélprogramként? Az eredeti XCoffee helyére egy weboldal került; nincs már különleges ügyfélprogram, elérhető X-nélküli ügyfelekről is, és az összes, nehézkes hálózatkezelési feladatot egy webkiszolgáló látja el. Ha tudomásod lenne egy olyan általános megoldásról, amely dédelgetett gyermekedet, a saját programodat, kisebbre varázsolhatná – felhasználnád?

Az olvasáshoz visszatérve: nézzük meg, mit is találunk ebben a számban! Ha ellenőrzést kívánsz gyakorolni a forráskód felett, esetleg olyan tervezetek fejlesztői kódját szeretnéd átböngészni, amelyek hivatalosan még nem jelentek meg, akkor szükséged lesz a CVS ismeretére. Az alapokat *Ralph Krause* ismerteti a 29. oldalon.

Számos gondot okozhatnak a memóriakezelési hibák: akár szakaszolási hibát (segmentation fault) találunk első C programunkban, akár olyan biztonsági hiányosságokat, amelyek



teljes céget dönthetnek romba. Memóriakezeléssel kapcsolatos tudás-

sodat gyarapíthatod, ha olvasol néhány szabadon elérhető, a GNU C könyvtárban található eszközről. *Petr Sorfa* a memóriakezelésről és a hibák elhárításáról szól a 32. oldalon. Ha cserzett arcú, öreg Unix-motoros vagy, aki az első kiadás óta velünk tart, talán kicsit gunyoros arckifejezéssel szemléled, ha az összetett fejlesztői környezetekkel kapcsolatban ejtünk szót valamiről. Míg Te ezen élcelődsz, addig mi mesélünk egy kicsit az olvasóknak a KDevelopról és a Glade-ről. A 36. oldalon *Mitch Chapman* ismerteti a Glade-et, azt a gyorsan megkedvelhető eszközt, amely segít elkülöníteni a GUI tervezését a kódolástól. Csak mutatni kell és kattintani, és máris kész a GTK-felület – a kódot pedig külön is meg lehet írni. A legjobb az egészben az, hogy Pythont használ a példákhoz, mely remekül illeszkedik abba az újfajta szemléletbe, amely szerint „használj új, objektumközpontú parancsfájlkészítő nyelvet, és nem kell többé a sebességgel foglalkoznod”.

A munkaasztalok harcából nem maradhat ki a KDE sem, *Petr Sorfa* cikke a KDevelopról az 76. oldalon olvasható, Mustra rovatunkban. Természetesen ahogy a KDE-alkalmazások is elsősorban a C++-ra és a Qt-re összepontosítanak, egy C++ programozó számára a Qt GUI kipróbálása egyet jelent annak megkedvelésével. A Qt esztétikai letisztultságát C++ programozó nehezen fejezheti ki szavakkal, és ez így is van rendjén. A KDevelop együttműködik a CVS-sel és a Qt Designerrel, amely – mint sejtethető, felhasználói felülettervező eszköz a Qt-hez; a Qt-ről bővebben majd a következő szám egyik cikkében szólunk. Sok örömet kívánok ezekhez a fejlesztői eszközökhöz, és ne feledd: nem árt néha abbahagyni a kódolást, és olvasgatni egy kicsit!

Kapcsolódó címek

Brian Behlendorf The World Needs Less Software (A világnak kevesebb programra van szüksége) ➔ [http://discuss.userland.com/msgReader\\$16542#16601Subversion](http://discuss.userland.com/msgReader$16542#16601Subversion)
 ➔ <http://subversion.tigris.org/>
 The Trojan Room Coffee Pot ➔ <http://www.cl.cam.ac.uk/coffee/qsf/coffee.html>
 The Art of Unix Programming (A Unix-programozás művészete)
 ➔ <http://www.tuxedo.org/~esr/writings/taoup/index.html>



Don Marti
 a Linux Journal szakmai szerkesztője, olvasóink a dmarti@linuxjournal.com címen érhetik el.