

A nyílt forráskódú szoftverfejlesztés társadalmi és ideológiai háttere

Az infokommunikációs technológiák fejlődésében nem pusztán technika fejlesztések szerepét, trendjeit érdemes vizsgálni, hanem azok létrejöttének társadalmi, ideológiai környezetét – a különböző értékeket, normákat, elképzeléseket. Különösen jó példája lehet ezen elképzeléseknek a szabad és nyílt forráskódú szoftverfejlesztés, melyet már kezdetétől döntően befolyásol a fejlesztésének társas környezete. Az alábbi írásban – elsősorban az Ubuntu Linux disztribúció példáján keresztül – arra igyekszem rámutatni, milyen társadalmi hatások, víziók, elvárások álltak és állnak a nyílt forráskódú fejlesztés hátterében, s ezeknek milyen hatásuk van a technológiai fejlesztésekre, illetve a közösség elképzelésére? A cikkben a Linux kialakulását lehetővé tévő domináns társadalmi ideológiák és professzionális praxisok 1960-as évekbeli kialakulásától napjaink változásaiig igyekszem áttekinteni a technológiai innováció mögött álló társadalmi hátteret.

Kulcsszavak: *nyílt forráskód, ideológia, hacker kultúra, információs technológia*

Szerzői információ:

Mátyus Imre a Szegedi Tudományegyetem Kommunikáció- és Médiatudományi Tanszékének tanársegéde, az SZTE Digitális Kultúra és Elméletek Kutatócsoport tagja. 2005-ben végzett kommunikáció szakos bölcsészként az SZTE-n, majd a Pécsi Tudományegyetem Kultúratudományi Doktori Programjának hallgatója lett. Fő kutatási területe az egyének és közösségek online identitásrepresentációjának vizsgálata, különös tekintettel a nyílt forráskódú fejlesztői és felhasználói közösségekre. E-mail: matyusimre@gmail.com

Így hivatkozzon erre a cikkre:

Mátyus Imre. „A nyílt forráskódú szoftverfejlesztés társadalmi és ideológiai háttere”.

Információs Társadalom XIV, 4. szám (2014): 62–83.

<https://dx.doi.org/10.22503/inftars.XIV.2014.4.4>

A folyóiratban közölt művek

a Creative Commons Nevezd meg! – Ne add el! – Így add tovább! 4.0

Nemzetközi Licenc feltételeinek megfelelően használhatók.

Mátyus Imre

A nyílt forráskódú szoftverfejlesztés társadalmi és ideológiai háttere

Bevezetés

A nyílt forráskódú szoftverfejlesztés kialakulása az 1990-es évek egyik legfontosabb fejleménye volt az információtechnológiai (IT) szektorban, melynek hatásai mára jóval túlmutatnak az infokommunikációs technológiák szegmensén. Az egyenrangú, sok esetben nem-professzionális felek önszervező közössége által megvalósított termelés (*peer production*) egyik legékesebb példája a mai napig a Linux operációs rendszerek fejlesztése. Smajda (2011) szerint a nyílt forráskód (a felhasználók számára szabadon alakítható és felhasználható szoftver, mely többnyire szabadon terjeszthető) az egyenlő felek közötti együttműködés elképzelése mára a társas élet számos egyéb területére kiterjedt a hadviseléstől a várostervezésen át a gyógyászatig. Az ideológiai alapok, etikai keretek a mai napig komolyan befolyásolják a nyílt forráskódú fejlesztések irányát. Ahogyan arra többek mellett Stewart és Gosain (2006) is rámutatott, az ideológia, a bizalom, illetve a kommunikáció kulcsfontosságú tényező ezen projektek sikeressége szempontjából.

Jelen írásban arra vállalkozom, hogy áttekintsem azon ideológiai és etikai alapokat, melyek egyaránt befolyásolták a tágabb IT-szektor, illetve a szabad és nyílt forráskódú szoftverfejlesztés kialakulását és működését. A jelenlegi cikkben egy sajátos példán, az Ubuntu Linuxon keresztül azt is szemléltetni kívánom, hogy milyen változások történtek, illetve történnek a nyílt forráskódú szoftverfejlesztés terén. Az infokommunikációs technológiák társadalmi beágyazottságának, használatának megváltozása ugyanis mára a korábban elsősorban a szakértő felhasználók szűk köre számára készített alkalmazások, rendszerek terén is érezteti hatását. Az Ubuntu által képviselt üzleti szemlélet pedig sok esetben feszültségforrássá válik, mivel a közösség érdekeivel, valamint a hackerkultúra ideológiai örökségével helyezkedik szembe.

Az alábbiakban elsőként a hackerkultúra bemutatásával, illetve a Himanen (2001) és Castells (2002) által hackeretikának nevezett érték- és normarendszer leírásával foglalkozom, melyek a mai napig meghatározzák a nyílt forráskódú fejlesztés, illetve a fejlesztő és felhasználó közösségek ideológiai hátterét. Mint látható lesz, a hackerkultúra fogalma mind Himanen, mind Castells esetében összemosódik a szabad és nyílt forráskódú fejlesztői közösségekkel.

1. A hackerkultúra mint ideológiai előfutár

A mindennapi diskurzusokban a hacker pontos fogalmi meghatározása sok esetben a médiatapasztalatokból felépített számítógépes alvilághoz és illegális praxisokhoz kapcsolódik. Az eredeti (értsd: a hackerkultúra történetében visszakövethető, az 1960-as évek programozói kultúrájából kinövő) definíciók viszont a hackert alapvetően nyitott, kreatív, kíváncsi individuumként igyekeznek meghatározni, aki meghatározott irányelvek és értékrend (a hackeretika) alapján cselekszik. Ezen feszültség részben annak tudható be, hogy a hackerkultúra értelmezése valójában csak saját belső etikájának keretei között értékelhető neutrálisnak – a kívülállók számára a hackerkultúra számos praxisa erkölcsi, vagy jogi szempontból megkérdőjelezhető. Alábbiakban elsősorban a hagyományos hackeretika Himanen (2001) és Castells (2002) által leírt tartalmát, gyökereit, és annak a nyílt forráskódú szoftverfejlesztésben megjelenő alkalmazását, hatását mutatom be, különös tekintettel az Ubuntu Linux disztribúcióra.

A fentebb vázolt külső (étikus) és belső (émikus) értelmezések közötti eltérésekből adódó problémák elkerülése érdekében az alábbiakban a hackerkultúrával kapcsolatban Manuel Castells leszűkített definícióját alkalmazom:

„[...] a hackerkultúra véleményem szerint arra az érték- és nézetrendszerre vonatkozik, amely a kreatív programozás önmeghatározott projektjeiben együttműködő, egymással online kapcsolatot tartó számítógép-programozók hálózataiból bontakozott ki” (Castells 2002: 51).

Castells meghatározása alapvetően Levy (2010 [1984]) történeti áttekintéséből merít, mégis, a fenti definíció elsősorban a nyílt forráskódú, önszervező szoftverfejlesztő projektek tagjaira alkalmazza a hacker kifejezést. Ennek legegkleltársabb példájaként maga a szerző is az alábbi cikk középpontjában álló Linux operációs rendszerek fejlesztésén dolgozó programozókat érti. Természetesen a hacker kifejezésnek létezik ennél jóval kevésbé leszűkített megközelítése is, amely kiterjeszthető a számítástechnikán túl számos területre – például a tudomány vagy a művészet terére (Himanen 2001, Castells 2002, Raymond 2007).

A következőkben elsősorban Levy (2010) és Raymond (2007) írásai alapján, röviden áttekintem a hackerkultúra 1950-es évektől napjainkig tartó történetét. E visszatekintés nem kizárólag a technológiai tényezők változására vagy az infokommunikációs eszközök fejlődésére összpontosít, hanem sokkal inkább a társadalmi kontextus változására, melyben a hackerkultúra igyekszik meghatározni önmagát, vagy éppenséggel külső értelmezésekben igyekeznek valamilyen viszonyt leírni a hackerek és a társadalom többi része között. Ez a dialektikus viszony a társadalom és a hackerek közössége között biztosítja a hacker identitás kialakulását és változását. S bár a médiareprezentációk sok esetben leegyszerűsített, erkölcsi, vagy jogi normák szemszögéből igyekeznek bemutatni a hackerkultúrát, ezen törekvések semmiképpen nem hagyhatók figyelmen kívül, hiszen esetenként ezek fényében alakítja (át), határozza meg saját magát a hackerek közössége.

Ezt követően a hacker etika leírásainak áttekintésére törekszem – különös tekintettel Levy (2010 [1984]), Himanen (2001), illetve Castells (2002) munkáira. Ezen külső

perspektívákból született interpretációk – melyeket igyekszem Raymond (2007) belső értelmezéseivel összevetve bemutatni – jobbára arra törekszenek, hogy a hackerek praxisainak hátterében álló elképzelések, szokások, értékek, normák alapján határozzák meg az adott kultúra határait, működésének alapjait.

A szöveg harmadik nagyobb egysége már elsősorban a Himanen (2001) és Castells (2002) által a hackerkultúrával „összeolvasott” nyílt forráskódú fejlesztő közösségek általános bemutatását nyújtja. Ebben a részben fejtem ki azt is, miben különbözik a szabad szoftver (free software) és a nyílt forráskódú szoftver (open source software). Utóbbi megkülönböztetés a későbbiekben is különösen fontos lesz, tekintettel arra, hogy az elnevezésbeli különbségek hátterében olyan ideológiai és gazdasági jellemzők állnak, melyek a hackeretikával, a hackerkultúrával kapcsolatos örökség különböző értelmezéseiből, illetve implementációjából fakadnak.

Fontos még leszögezni, hogy a hackerkultúra kifejezést az alábbiakban – Castells (2002) nyomán – tudatosan arra az érték-, norma- és szokásrendszerre, valamint az ezek mögött álló tudásokra, meggyőződésekre alkalmazom, melyek általánosan jellemezhetnek bizonyos¹ programozói közösségeket. Maguk a hackerközösségek a többségi társadalommal való viszonyban inkább szubkultúráként vagy csoportkultúráként határozhatók meg.

1.1 A hackerkultúra korai története

A hackerek korai generációinak történetét Steven Levy (2010 [1984]) és Eric S. Raymond (2004) is összefoglalja. Érdemes kiemelni, hogy mindkét szerző szövegeiben gyakran találkozunk epikus elbeszélő stílussal², illetve vallási vagy mágjával kapcsolatos metaforákkal, kifejezésekkel. Ez egyrészt a hackerkultúra játékoságából eredeztethető, másrészt a saját kultúra misztifikálásának, kiemelésének, a kívülről, hétköznapi társadalomtól való megkülönböztető gyakorlatként működik. Ez a fajta belső zsargon³ segíti a csoportkohézió kialakítását, illetve megtartását – a beavatott és a kívülről elválasztását is (ugyanakkor a megfelelő kifejezések ismerete nem helyettesíti a tudást és a teljesítményt).

A hacker kifejezés az angol „to hack” igéből származik, melynek eredeti jelentése „durván ácsolni”. A hacker terminust először a Massachusetts Institute of Technology (MIT) az 1946/1947-es tanévben létrehozott Vasútmodellező Klub (Tech Model Railroad Club, TMRC) tagjai kezdték egymásra alkalmazni. Az MIT TMRC afféle keveréke volt az 1920-as években kialakult elektronikus barkácskultúrának, illetve a tudományos élet innovatív hozzáállásának. A klub tagjai próbáltak olyan elmés, ugyanakkor nem

¹ A konkretizálás hiánya e helyütt szándékos – a meghatározás pontosabb kifejtésére a hacker és a hackerkultúra pontosabb definiálása kapcsán térek rá.

² Raymond *A hackerség rövid története* című esszéjében egyenesen legendákkal és csodákkal teli eredettörténetként írja le a hacker kultúra történetét 1945-től az 1990-es évek derekáig.

³ A zsargon terminus használata e helyütt azért is indokolt, mert a hackerkultúra egyik legfontosabb alapinformáció gyűjteménye, a hackerek online „szótára” maga is a *The Jargon File* néven fut (lásd: <http://catb.org/jargon/>).

feltétlenül esztétikus technikai megoldásokat („hack”-eket) kieszelni, amivel jobban működő vasútmodelleket (távlati szempontból pedig valódi vonatokat) lehet készíteni. A vasútmodellezők között voltak az MIT mesterségesintelligencia-kutató részlegének későbbi tagjai is, akik aztán saját programozói köreikben kezdték tovább használni a kifejezést. Az 1960-as években így az MIT-n belül a hacker kifejezés beszivárgott a számítástechnika világába, ahol elsősorban olyan programozókra kezdték alkalmazni, akik valamilyen okos megoldással optimálisabb teljesítményre, vagy valamilyen új feladat megoldására tudták rávenni a rendelkezésükre álló eszközöket.

A Raymond (2004) által *korai hackerkultúrának* nevezett időszak másik fontos helyszíne az MIT mellett a Xerox Palo Alto Kutató Központja (Xerox PARC) volt, ahol az 1980-as éveket megelőzően a vállalati számítástechnika területén jelentős szerepet betöltő cég fiatal kutatókat foglalkoztatott elsődlegesen azzal a céllal, hogy minél érdekesebb és hasznosabb újításokkal bővítsék a cég repertoárját. A Xerox ugyanakkor számos újításban (például. az eger és a grafikus felhasználói felület kombinációja, a belső hálózat kiépítése, a lézernyomtató) nem látta meg időben az üzleti potenciált, ezért ezen innovációk gyakran más cégeken keresztül kerültek piacra.⁴

Ezt a korszakot elsősorban a kevés, nagy teljesítményű terminál használata jellemezte, melyek típusonként más operációs rendszert használtak, ezért az egyes rendszerek (számítógéptípusok) közötti mozgáshoz komoly plusz ismeretekre volt szükség. 1969-ben kezdődött meg az első architektúrafüggetlen (azaz nem egy adott géptípushoz kötődő), hordozható, a korábbinál egyszerűbb szerkezetű operációs rendszer, a Unix fejlesztése. Az AT&T Bell Laboratories keretei között elindított fejlesztés az 1980-as évek elejéig szabadon (az AT&T beavatkozása nélkül) folyt, és a Unix közkedvelt operációs rendszer lett a különböző egyetemeken. A Unix fejlesztése és terjesztése mindaddig viszonylag kötöttségmentes volt, míg az AT&T meg nem sejtette a mögötte rejlő üzleti lehetőséget, és magának nem követelte a fejlesztés jogát („lezárva” ezzel a korábban nyíltan fejleszthető forráskódot), mely lépés számos fontos következménnyel járt a hackerkultúrában.

Ugyancsak az 1960-as és 1970-es évekhez kapcsolódik az internet elődjeként számmon tartott, katonai költségvetésből indított ARPAnet kialakulása is. Az 1969-ben elindult rendszert – bár tagadhatatlanul volt katonai jelentősége is – elsősorban mégis az egyetemek és kutatóközpontok számára lényeges számítógép-kapacitás növelésére, illetve a tudományos eredmények könnyű és gyors megosztására használták. Az ARPAnet így a MilNet katonai hálózat 1983-as leválásáig tulajdonképpen elsősorban a civil kutatói szféra számára volt értékes (Castells 2002).

Az egyetemek és kutatóintézetek közege különösen nagy hatással volt a korai hackerkultúra alakulására. Ahogyan a későbbiekben a hackeretika tárgyalása kapcsán rámutatok, az olyan akadémiai örökségek, mint a munka öröme, a folyamatos kíváncsiság

⁴Az egyik jó példa erre a Xerox PARC-nál kifejlesztett, egeret használó (ablakokat és ikonokat alkalmazó) grafikus felhasználói felület volt, melyben a cég nem látta meg időben a kínálókozó lehetőségeket, és tulajdonképpen átadta ötleteit az Apple-nek, akik később ezzel teremtették meg a felhasználóbarát személyi számítástechnika alapjait. A grafikus felület és az eger ötletét aztán sajátos módon a Microsoft „kölszönözte” tovább az Apple-től.

vagy a teljesítményközpontú alapelvek visszaköszönnek az eredeti hackerkultúrában is (Himanen 2001, Castells 2002, Raymond 2004).

A XX. század második felének libertáriánus eszméi, melyek az egyéni szabadság felé gravitáltak (és sok esetben szintén az egyetemi élet tereiben érintkeztek a bimbózó hálózati kultúrával) szintén fontos szerepet játszottak a hackerkultúra kialakulásában. Himanen (2001) hackeretikáról írott könyvének epilógusában Castells is rámutat, hogy az egyre nagyobb területeket behálózó infokommunikációs hálózatok kialakulása, illetve a kapitalizmus és az államiság gazdasági/politikai krízisei mellett az 1960-as, 1970-es évek libertáriánus mozgalmainak is erőteljes hatása volt a kortárs társadalom alakításában, elsősorban a szabadság kérdésének középpontba emelésével, illetve a létező intézményekkel szemben megfogalmazott kritikáikkal. A hackeretikában különösen jól érzékelhető a libertárius szocializmus politikai filozófiájának nyoma, amely az anyagi (és információs) javak egyenlő elosztására, a magántulajdon felszámolására, az egyéni önrendelkezésre, valamint a hierarchia- és bürokráciamentes társadalom kialakítására fekteti a hangsúlyt. Ezen elképzelések köszönnek vissza a hackerkultúra ajándékkultúrájában, szabadságba vetett hitében, illetve a magántulajdonhoz (legyen szó akár kereskedelmi szoftverről vagy korlátozott hozzáférésű számítógépes hálózatról) való viszonyában, illetve az intézményi hierarchiából származó tekintéllyel kapcsolatos hozzáállásában.

Azt is fontos megjegyezni, hogy az 1980-as éveket megelőzően a legtöbb szoftver viszonylag kötetlenül terjedt, illetve az egyéni igényekhez igazodó módosításuk nem vont maga után jogi felelősséget. A programozókat nem kész szoftverek, hanem ledolgozott munkaórák után fizették, és nem befolyásolta munkájukat, hogy a kezük közül kikerülő alkalmazások miként terjednek tovább. Ennek oka részben az volt, hogy az információtechnológia vezető cégei (például az IBM vagy a Xerox) elsősorban a gépek (a hardver) és nem az alkalmazások (a szoftver) eladásában látott profitot. Az 1980-as évek első felében azonban elsősorban a személyi számítógépek sikeréből fakadóan a cégek hozzáállása megváltozott e tekintetben, ami a szabad és nyílt forráskódú szoftverfejlesztés kialakulásához vezetett.

1975-től a hackerkultúrában új elem jelent meg: az egyre növekvő teljesítményű személyi számítógépek, és a telefonos hálózatok segítségével összekapcsolt felhasználók fiatal hackergenerációja. Raymond (2004) szerint így alapvetően három eltérő gyökerű és technológiai hátterű réteg ért össze a hackerkultúrában: az MIT és a Xerox „nagy öregjei”, az egyetemeken Unixon szocializálódott programozói és a hálózatiság, a közösség erőnyelveit kiemelő személyiszámítógép-felhasználók. Utóbbi réteg teremti meg a hackerkultúrában a kollaboratív fejlesztés hagyományát, amely a jóval zártabb rendszerekben (az MIT-n használt terminálok vagy a Unixot futtató gépeket összekapcsoló hálózatokban⁵) nem tudott kibontakozni (Raymond 2004).

Smajda (2011) megfigyelésében az 1980-as évek derekára kialakultak annak az „erkölcsei mezőnek” az alapjai, melyben a hackerkultúra, illetve a személyi számítástech-

⁵ Az ARPAnet, majd az internet csomópontjai közötti kommunikációt a TCP/IP protokoll szabályozza, míg például a Unixot használó nagy számítógépek (mainframe-ek) egy saját UUCP-protokollal csatlakoztak egymáshoz. Ebből fakadóan a két protokoll közötti illeszthetőség hiánya miatt a Unixot használó gépek egy kisebb, zártabb hálózatban kapcsolódtak egymáshoz, míg a TCP/IP protokollt használó személyi számítógépek kapcsolódási köre folyamatosan bővült az internet fejlesztésével.

nika kezdeteinek fontosabb szereplői mozogtak. Smajda Bourdieu „mező” fogalmát használja, mely saját összefoglalásában olyan „kapcsolati teret jelent, melyben az egyes ágensek pozícióját a többiekhez való viszonyuk jelöli ki, és amelyben a közös, hallgatólagosan elfogadott mezőbeli szabályok határozzák meg az egyéni hiteket, motivációkat és feltevéseket” (Smajda 2011: 307). A korai számítástechnika erkölcsi mezeje alapvető elképzeléseit illetően ugyan egységesnek tűnt, amennyiben a számítógépet afféle demokratizáló eszköznek tekintette, normák és motivációk terén azonban már akkor számtalan eltérés mutatkozott az egyes szereplők között. Az eltérő megközelítések pedig a későbbiekben eltérő fejlesztési modellek megszületését eredményezték.

Az 1980-as évek két fontosabb fordulatot is jelentett a hackerkultúra életében. Egyrészt a korábban neutrális hacker karaktere a különböző médiareprezentációkban negatív színt kapott, illetve a zárt forrású fejlesztések beindulása elindította a szabad és nyílt forráskódú fejlesztés fejlődését. Richard Stallman 1982-ben jelentette be, hogy egy új, a Unix-hoz hasonló, de attól teljesen független, szabadon módosítható és terjeszthető operációs rendszer készítésébe kezd. Ez volt a GNU-projekt, mely a szabad szoftverek első markáns képviselője volt. 1984-ben jelentette be az AT&T, hogy mégis igény tart a Unix operációs rendszerre, így a korábbi nyílt fejlesztésből kisorult programozók új Unix-klónt indítottak BSD (Berkley Software Distribution) név alatt.

Az 1990-es évektől a hackerkultúra elindult a professzionalizálódás útján: a korábban inkább a szabadidőben folytatott programozói, fejlesztői praxisok a megélhetés eszközkövé kezdtek alakulni. Ez figyelhető meg a későbbiekben elemzett szabad szoftverfejlesztés és nyílt forráskódú fejlesztés szétválásában is. Szintén fontos lépcső volt a Linux fejlesztésének, majd ennek folyamányaként a Nyílt Forráskód Kezdeményezésnek (Open Source Initiative) az elindítása. Thomas (2004) úgy véli, hogy az 1990-es évek legfontosabb változása a hackerkultúra szempontjából éppen két operációs rendszer, a Windows és a Linux megjelenéséhez kapcsolódik, melyekhez eltérő fejlesztői és felhasználói szemléletek kapcsolódnak. A két operációs rendszer szembeállítására szimbolikus: míg a Windows a zárt fejlesztés, a profitszerzés és a felhasználóbarát termék megtestesítője, addig a Linux a nyílt kollaboráció, a transzparencia élővása, melynek használata ugyanakkor nagyobb technológiai hozzáértést igényel. A két fejlesztési modell szembenállása (melyeket Raymond a „katedrális” és a „bazár”⁶ metaforájával ír le) a hackerkultúra – különösen pedig a Linux fejlesztői közösségeinek – fontos identitás-szervező elemévé vált.

A hackerkultúra részletesebb leírásai nagyjából az 1990-es évek derekáig követik a történeteket. Mára a hackerek közösségei, a fejlesztések, a technológiai változások olyan mértékben megnöttek, hogy az átfogó kép kialakítása közel lehetetlen. A hackerkultúrában korábban kimutatható nagyobb rétegek helyett számos kisebb szubkultúra alakult ki (egyes Linux-disztribúciók vagy programnyelvek, mint a Python vagy a Perl közösségei). A professzionalizálódást elősegítő legitimációs törekvések – melyek a hackerek

⁶ Raymond (2004) a zárt forráskódú fejlesztést a katedrálisépítéshez hasonlítja: kiváltságos hozzáértők egy kis csoportja a társadalom többi részétől elszigetelten megtervezi és megvalósítja elképzeléseit, melyet azok már csak kész állapotában láthatnak meg. A tervezés és a fejlesztés folyamata maga el van zárva kívülállók szemei elől. A bazár (nyílt forráskódú) modell ezzel szemben nyitott, aktív együttműködésen alapul, melyet folyamatosan figyelemmel lehet kísérni, és be lehet kapcsolódni az egyes feladatokba.

hozzáértését, technikátörténeti szerepét kihangsúlyozva igyekeztek gazdasági szerep-növekedést elérni – csak részben voltak eredményesek. A média által kiemelt újabb és újabb incidensek, a hacktivizmus⁷ felélénkülése továbbra sem állította helyre a hackerkultúra megítélését. Habár a korai vagy eredeti hackerkultúra, mely a technológiai fejlődés, a szabadság és nyitottság eszméi köré szerveződik alapvetően hozzájárulhat a társadalmi problémák (például a digitális szakadék, a szélsőséges társadalmi rétegződés) felszámolásához, a hackerekről kialakult negatív kép komolyan visszavetette e kultúra megítélését. Az illegális praxisok előtérbe kerülése a többségi társadalom szemében a hagyományos hackerkultúrához tartozókat marginalizálta, atipikussá tette saját kultúrájukban (Nissenbaum 2004).

1.2. Hackeretika

A hackeridentitás, illetve a hackerkultúra pontos meghatározása szempontjából kiemelt jelentősége van a hackeretikának, mely afféle gravitációs pontja lehet a nyílt forráskódú fejlesztői közösségek értelmezésének. Természetesen nem hagyhatjuk figyelmen kívül azokat az aggályokat, melyeket a hackerkultúra belső heterogenitása kapcsán – az egyes szubkultúrák, illetve a hackerek és crackerek közötti határvonalak „szokatlan képlékenység” (Taylor 1999: vii) –, vagy az egyes hackerek saját hitvallásai és a többség praxisai közötti különbségek kapcsán (Lin 2007) fogalmazhatóak meg. Most elsősorban arra keresem a választ, hogy a Levy (2010), Himanen (2001), illetve Castells (2002) által felvázolt hackeretika nyomai megtalálhatók-e a jelenlegi nyílt forráskódú szoftverfejlesztői közösségek önreprezentációjában, illetve önértelmezésében? Lin (2007) Adrian MacKenzie kételyeit idézi:

A hacker hősök egyéni etikai értékeinek a hackerkultúra magjaként való közép-pontba állításával nagyban figyelmen kívül hagyjuk a szoftverfejlesztés bonyolult praxisait valami olyasmi kedvéért, amit én csak szenvedélyközpontú individualizmusként tudok értelmezni (...) (MacKenzie 2001: 544, idézi Lin 2007: 33).

Lin arra is felhívja a figyelmet, hogy a fejlesztők nem feltétlenül egy adott vízió mellett kötelezik el magukat, hanem egy adott fejlesztési projektben való részvétel *által* válnak bizonyos etikai elképzelések megvalósítójává. Mindezeket szem előtt tartva hozzátehetjük, hogy a vázolt egyéni víziók, etikai irányelvek mégiscsak származnak „valahonnan”. Nem kívánom vitatni azon alapkérdéseket, hogy vajon a fejlesztői praxis, a projektek szerveződése, működése hoz létre valamiféle etikaként feltételezhető mögöttes szabályrendszert, vagy valamilyen meglévő etikai elvek szervezik magát az együttműködést. Alábbiakban mindössze arra törekszem, hogy a fentebb említett szerzők által vázolt hackeretika körvonalát felvázoljam.

⁷ A hacktivizmus alapvetően valamilyen társadalmi vagy politikai céllal összefüggő aktivizmus, mely azonban a hackerek (illetve sok esetben a crackerek) technikáit (rendszerek feltörése és tartalmak megváltoztatása, kiszivárogtatása) alkalmazza.

A hackerkultúrával kapcsolatos alapvető jellemzőket az elsők között Levy (2011 [1984]) rögzítette. Az MIT hackereinek közös értékrendjében központi szerepet játszott a szabadság, az intelligencia, a technikai hozzáértés, valamint a számítógépben rejlő lehetőségek kibontakoztatása iránti érdeklődés. Ezen alapvető vonásokból Levy (2010: 28–38) hat konkrét szabályon keresztül mutatja be a hackeretika lényegét:

1. A számítógépekhez (illetve bármely más eszközhöz, melyek segítenek megismerni a világ működését) való hozzáférésnek teljes mértékben szabadnak kell lennie!
2. Minden információt szabadon hozzáférhetővé kell tenni!
3. A hackernek nem szabad megbízni a hatalomban, és a decentralizációt kell támogatnia!
4. A hackereket csak és kizárólag hackeri tevékenységük alapján szabad megítélni, nem pedig más, álságos kritérium (például kor, nem, etnikai hovatartozás, titulus) alapján!
5. A számítógép alkalmas a művészet, a szépség létrehozására.
6. A számítógépek jobbá tehetik az életet.

E rövid kivonatban jól tetten érhetők azon elemek, melyeket nemcsak Levy, hanem a későbbiekben részben épp az ő leírásaiból merítő Himanen (2001), Castells (2004), illetve Raymond (2007) is hangsúlyoz. A hacker számára a számítógép – praktikus felhasználási lehetőségein felül – az információhoz, a tudáshoz, valamint az alkotáshoz fűződő szabadság szimbóluma.

Ahogy Levy megfogalmazza, „a forradalmi hackeretikának ezen elveit jobbra nem viták és beszélgetések alakították, hanem a programozók csendes egyetértése. Nem születtek kiáltványok. Nem voltak misszionáriusok, akik téríteni próbáltak volna” (Levy, 2010: 27). A Levy által vázolt hackeretika tehát inkább valamiféle közös tapasztalatból és értékrendből fakadó irányelvek összességeként fogható fel, mely az adott közösség – az MIT vasútmodellező klubjának – tagjai között különösebb formális rögzítés nélkül alakult ki, és hagyományozódott. Érdeemes felhívni a figyelmet arra, hogy mind Levy fogalomhasználatában, mind az általa felidézett programozói hitvallásokban, és példaként citált szövegekben⁸ megjelennek bizonyos, vallással (például: misszionárius, megtért [27], számítógép-vallás [170], messianisztikus cél [173]), illetve mágiával kapcsolatos (például: varázslók, boszorkányság) kifejezések. Ezek egyrészt elhatárolják az adott kultúrát a többségi társadalom, a kívülállók hétköznapi valóságától, s azt valamiféle misztikus, transzcendentális, hermetikus jelleggel ruházzák fel. Másrészt ezen nyelvi elemek kihangsúlyozzák azt a fajta érzelmi és ideológiai kötődést, melyet a hackerkultúra tagjai a közös élményből, a közös értékrendből – egyáltalán a közösséghez tartozásból

⁸Levy (2010) a Community Memory terminálok kapcsán felidézi Richard Gary Brautigan *All watched over by machines of loving grace* (Mindannyiunkat az isteni kegyelem gépei vigyázzák) című versét, mely a természet és a kibertér összeolvadásának vágyáról szól. E vers ihlette a Community Memory terminálok készítőit, hogy cégüket Loving Grace Cyberneticsnek nevezzék el. Ahogy Levy fogalmaz: „A Leopold’s-ban felállított gép nem pusztán terminál volt, hanem az Isteni Kegyelem eszköze. Az volt a szerepe, hogy a tudatlan nyáját a szeretetteljes hackeretika által termékkenné tett legelőkre terelgesse” (Levy, 2010: 175).

– merítenek. Nem utolsósorban pedig a hackerkultúra játékosága is megmutatkozik az ilyen megnyilatkozásokban.

Az elit közösséghez való tartozás érzésének erősítése annál is inkább helyénvalónak tűnik, mivel Levy leírásában a korai hackerkultúra tagjai (elsősorban az 1960-as, illetve 1970-es évek programozói) zömében saját korcsoportjaik által kivetett, esetenként lenézett „különcök”, „okostojások” voltak. Raymond (2004) értelmezésében a hackerrek elkülönülése a társadalom többi tagjától szükséges velejárója annak a problémafelismerő, racionalizáló attitűdnek, ami ugyan elengedhetetlen feltétele a hackerré válásnak, ugyanakkor megnehezíti a hétköznapi élethelyzetekben megjelenő irracionális elemek kezelését. Turkle (2005) arra mutat rá, hogy a hackerrek elkülönülése társaiktól nemcsak a szélesebb korcsoporton belül, hanem saját szűkebb érdeklődési területük, a számítástechnika terén is kialakul – az informatika iránt érdeklődők között is marginalizálódhatnak.

Castells (2002) maga is kiemeli, hogy a kutatóközpontok és egyetemek techno-meritokrata kultúrája képezte a hackerkultúra alapját. Ezt egyrészt alátámasztja az a tény, hogy a korai hackerkultúra valóban elsősorban a megfelelő infrastruktúrával rendelkező tudományos kutatóhelyeken (többek között az MIT-n és a Stanford Egyetemen) bontakozott ki. Másrészt a Castells által leírt techno-meritokrácia (vagy technoeit) számos jellemzője megtalálható a hackerkultúrában is. A legnyilvánvalóbb egyezés két belső jellemzőben mutatható ki: a kimagasló szakmai képességek és teljesítmények elismerésében, illetve a fejlődés, a tudás gyarapításának középpontba helyezésében. Ugyanakkor azt is hozzá kell tennünk, hogy míg a technoeit esetében a szakmai kiválóság valóban fontos presztízsbiztosító elem lehet az adott kultúra értékrendje szerint, a megbecsülést, az előbbre jutást ugyanakkor más intézményi és szervezeti kérdések, a hierarchikus intézményi keretből eredő kényszerek is befolyásolják. A hackerkultúrában Castells szerint a nyilvánvaló hozzáértés és a közösség számára hozzáférhetővé tett eredmények jelentik az adott kultúrán belüli presztízsszerzés kizárólagos lehetőségét.⁹ Az innováció, illetve az ismeretek bővítése mindkét kultúrában központi jelentőségű, s ennek folyamánaként az is kimutatható, hogy maga a fejlődés mind a technoeit, mind pedig a hackerrek számára önmagában értékes. Raymond (2007) beszámolója, illetve Kazári (2003) és Flammich (2002) interjúja alapján úgy tűnik, a számítógépes hálózatok biztonsági réseit vagy az alkalmazások hibáit kereső, képességeiket próbálgató hackerrek számára cselekedeteik megítélésében legalábbis részben szerepet játszik az a meglátás, miszerint a fejlődés nem szorítható morális vagy törvényi korlátok közé. (És arról sem szabad megfeledkeznünk, hogy a konvenciók és szabályok büntetlen áthágásából, illetve az izgalomból származó öröm legalább olyan fontos építőköve a hackerkultúrának, mint az alkotás öröme.) Ezen önkényes külső szabályozók a hackerkultúrában sok esetben a fejlődés, a tökéletesedés, a lehetséges potenciál kiteljesítésének gátjaiként jelennek meg, melye-

⁹Castells azon meglátása, miszerint a hackerkultúrában az intézményi keretek, a merev hierarchia hiányából fakadóan kizárólag az egyéni teljesítmény biztosítja a presztízst, elméletben mindaddig tartható, míg a háttérben finansziális kérdések nem vetődnek fel. Az olyan projektek esetében ugyanis, ahol a termelés támogatása részben, vagy teljes egészében gazdasági szervezetek kezében van, és a cél a profitszerzés, a szakmai kiválóság mellett más tényezők is befolyásolják a döntéseket.

ket egy fontosabb cél (legyen az a hacker saját fejlődése, vagy a szélesebb társadalom gyarapodása) át lehet, illetve át kell lépni.

Levy fent összefoglalt, előíró jellegű etikai kódjával szemben – mely inkább követendő viselkedési mintákat ír le – Himanen (2001) arra törekszik, hogy értelmezze és elhelyezze a hackeretika helyét a hálózati társadalomban. Továbbá, amíg Levy leírása elsősorban a hacker egyéni praxisainak, fejlődésének körülírását célozza meg, Himanen inkább arra helyezi a hangsúlyt, hogy a hackerkultúrában megjelenő elképzelések, normák, szokások hogyan illeszkednek a hálózati társadalom kontextusába, és milyen alapot nyújtanak egy lehetséges új termelési felfogáshoz.

Himanen (2001) könyve már címében – *A hacker etika és az információs kor szelleme* [*The Hacker Ethic and the Spirit of Information Age*] – is egyértelműen utal Max Webernek a protestáns etika és a kapitalizmus kapcsolatát tárgyaló művére (lásd: Weber 1995). Himanen arra törekszik, hogy a protestáns etikával összevetve bemutassa, milyen etikai alapokon működik az információs (hálózati) társadalom egyik legszignifikánsabb termelő kultúrája. Úgy véli, hogy az etikai alapelvek azonosításán keresztül megfelelően felvázolható azon társadalmi átalakulások halmaza, melyek az utóbbi fél évszázadban bekövetkeztek.

Castells a könyvhöz írott epilógusban is igyekszik szélesebb társadalmi kontextusba helyezni Himanen meglátásait. A hangsúlyt a technológia, illetve a technológiai paradigma változására helyezi „a kulturális, gazdasági, politikai és technológiai faktorok komplex interakciójában” (Himanen 2001: 155). Az általa *hálózati társadalom*nak nevezett (Castells 2005, 2006) kortárs társadalomban a vezető technológiai paradigma már nem a természeti erőforrások kiaknázására alapul (ahogyan az ipari társadalomban), hanem az információtechnológiai rendszerek által létrehozott *informacionalizmus* válik gazdasági szempontból a legpotenciálisabb termelő erővé. Castells hangsúlyozza, hogy az informacionalizmus „egy technológiai paradigma”, mely „nem társadalmi szervezeteit, és nem intézményeket takar (...), hanem alapot teremt egy bizonyos társadalmi szerkezethez” (Himanen 2001: 158), melyet ő hálózati társadalomnak nevez. Az informacionalizmus nem merül ki a tudás és információ szerepének felértékelődésében a gazdasági javak vagy a hatalom létrehozásában, sokkal inkább az információfeldolgozás változásának hatása válik fontossá a tudás előállításában és alkalmazásában. Egy ilyen technológiai környezet több ponton szükségszerűen átalakítja, befolyásolja a társadalmi berendezkedést a munka világától, a tudomány fejlődésén át, a gazdasági és politikai kapcsolatrendszeréig. Ebben a kontextusban pedig Castells fontosnak tartja azt a fajta új viszonyt a munka világával, amelyet a hackerkultúra képvisel.

Himanen (2010) a hackeretika kapcsán három szintet különböztet meg: a munka-etika, a pénz-etika, illetve a *netika* szintjeit. A munkaetika tárgyalása során arra mutat rá, hogy amíg a protestáns etika számára a munka az élet lényege, az egyén kötelessége, addig a hackerkultúrában a munka nem kizárólag önmagáért fontos, hanem azért, mert örömforrást jelent. Az odaadó programozás, a szenvedély realizálása a hackeretika egyik kulcspontja. Himanen szerint a hacker számára a munka egyszerre élvezetes hobbi és komoly termelőtevékenység. A munka élvezete ugyanakkor nem korlátozódik a hackerkultúrára – a hackerkultúra előzményeként szolgáló akadémiai kultúrában (és a művészetben) is jelen van ez a hajtóerő. Castells (2002) is válaszol az akadémiai kultúra (az ő megfogalmazásában a hálózati társadalomra vonatkoztatva techno-meritokrácia) és

a hackerkultúra összefüggéseit, de amíg Castells a különbségeket hangsúlyozza, addig Himanen a hasonlóságokat helyezi a fókuszba.

Himanenhez hasonlóan Castells (2002) is több helyen hangot ad azon meggyőződésének, hogy a hackerek kapcsolata saját érdeklődési, működési területükkel erős érzelmi kötődésen alapul. A kiemelt emocionális kapcsolódás elsősorban ugyan az eszközökkel végzett munka, a programozás, az alkotás kapcsán válik fontossá a hackeretikában, azonban a technológia birtoklása maga is örömforrásként értelmezhető – a számítógép az egyéni lehetőségek, tehetségek kibontakoztatásának eszközeként ugyanúgy működtethető, mint a külvilág problémái elől való menekülésre kialakított „biztonságos mikrovilágként” (Turkle 2005: 191).

Amíg a protestáns etikában az idő és a munka között egyfajta organikus kapcsolat volt leírható, és az időkomponens vált fontossá a mindennapi élet szervezésében, a hackerkultúra rugalmasabb időszemlélettel működik – az időfelhasználást igyekszik saját igényei szerint alakítani úgy, hogy a játékoság, a kikapcsolódás, a szenvedély kiélése helyet kapjon a mindennapi rutinban (Himanen 2001). A szenvedélyes munkavégzés ugyanakkor maga után vonja az algoritmizálható, repetitív feladatok megvetését (lásd: Raymond 2004).

A pénzetika kapcsán Himanen (2001) felveti, hogy amíg a protestáns etika a pénzt és a munkát egyaránt célként értékelte, a hackerek számára a munka inkább eszköz a pénzszerezésre. A pénz azonban nem önmagában való cél, hanem a finansziális függetlenség megteremtésének eszköze, melynek elérése megnöveli a saját szenvedély kiélésére fordítható idő mennyiségét. Emellett fontos kiemelni a hackerkultúrában az ajándékozás fontosságát. A nyílt forráskódú fejlesztők (szenvedélyüknek hódolva) idejük bizonyos részét, tudásukat, munkájukat a közösség javára fordítják, melyért cserébe megbecsülést kapnak. A hackerek ajándékkultúrája a reciprocitáson alapul – a közösség tagjai azért adnak „ajándékot” (időt, munkát) a közösségnek, mert úgy vélik, a közösség többi tagja is hasonlóan jár el (Castells 2002). Ráadásul a közösséggel megosztott fejlesztés a presztízsszerzés, az egyén kiemelkedésének, saját szakmai hozzáértésének kinyilatkoztatása. Raymond (2004) arra is rámutat, hogy a szakmai tekintélyből, illetve a közösségért végzett munkából származó hírnév a hackerkultúra piaci tevékenységgel érintkező szegmenseiben gazdasági haszonná konvertálható – a hírnév megbízásokat, munkaaajánlatokat eredményezhet. De az elismerés, az ajándékozás nem helyettesíti a szenvedélyt, hanem a szenvedélyes alkotás eredményeként jön létre (Himanen 2001). Az ajándékozás komoly presztízsvértéket képvisel a nyílt forráskódú fejlesztők köreiből, mely mind az egyén, mind a közösség számára fontos erőforrást jelent (Bergquist és Ljungberg 2001). Előbbi számára szakmai elismerést, fejlődést jelenthet, utóbbi számára afféle összetartó erőt.

„Ez a kultúra úgy is felfogható, mint a kollektívizmus és az individualizmus valamiféle keveréke: a közösségnek nyújtott ajándék teszi az egyént hőssé a többiek szemében. A hősök nagy hatású példaképek, ugyanakkor erős vezetők is lehetnek. Jelenlétük fontos, mert általuk válnak ezen heterogén környezet eltérő értékei és céljai egységessé, koncentrálnak a közös célra, ugyanakkor legitimálják a hatalmi viszonyok asszimmetriáját is” (Bergquist & Ljungberg 2001: 319).

Az ajándékozással szerzett presztízs fentiek alapján nem pusztán a közös érték- és normarendszer egyik fontos jellemzőjeként, de a kultúrán belüli hatalmi struktúra fenntartójaként is kiemelt szerepet játszik.

A hackeretika harmadik, sajátos szintje a Himanen (2001) által *netikának* (hálózati etika) nevezett réteg, melynek lényege az információk szabad áramlásába, a közösség (és szélesebb értelemben az emberiség) fejlődésébe vetett hit. A nyílt forráskódú fejlesztés hátterében olyan nyitott, inkluzív közösségek állnak, amelyek a hálózati erőforrások megfelelő szervezésével és menedzselésével nem csupán közös diskurzusokban vesznek részt, hanem értéket termelnek. A nyitottság hangsúlyozása, a technológiai lehetőségek messzemenő kihasználása mellett a hackerkultúra nagy hangsúlyt fektet például a személyes adatok, a magánszféra védelmére, s bizonyos tekintetben saját felelősségének tekinti a hálózati biztonság fejlesztését (amely törekvésnek jó példája az etikus hackerek megjelenése).

S bár a nyitottság, a véleménynyilvánítás és az alkotás szabadsága önmagában is érték a hackerkultúra számára, nem szabad megfeledkeznünk e helyütt sem az akadémiai gyökerekről. A tudományos életben a fejlődés záloga az új kutatási eredmények minél gyorsabb és hatékonyabb megosztása – ez az elképzelés pedig ugyanúgy fellelhető a hackerkultúrában, amely az információk, eredmények, fejlesztések megosztásában a közösség fejlődésének kulcsát látja (Castells 2002). Az eredmények, az innováció, a létrehozott érték a hackerkultúrában a közösségnek nyújtott ajándékként kerül megosztásra. A cserekultúra és az ajándékkultúra összevetése kapcsán elmondható, hogy a bőségen alapuló ajándékozó kultúra sokkal inkább képes az információk javak felhalmozására, mintsem a hiányon alapuló cserekultúra. S míg utóbbiban az képvisel értéket, presztízt, amit birtoklunk (amit el lehet cserélni), addig az ajándékozás kultúrájában a megbecsülés ahhoz kapcsolódik, amit megosztunk a közösséggel (Raymond 2004).

Fentiekből nagy vonalakban kirajzolódik egy olyan etikai háttér, mely a nyílt forráskódú szoftverfejlesztő projektek működésének hátterében állhat. Ezen irányelvek, meggyőződések nincsenek szigorúan kodifikálva a hackerkultúrán belül, hatásuk mégis érezhető. A nyílt forráskódú fejlesztési projektekben létezik valamiféle szokásjog, és léteznek tabuk (például a projektek tulajdonjogával, a módosítások terjesztésével vagy az elágaztatások¹⁰ indításával kapcsolatban), melyeket a közösség tagjai anélkül is szem előtt tartanak, hogy azoknak pontosan tudatában lennének. Ez alapján elmondható, hogy vannak olyan erős viselkedési minták, célkitűzések, normák, melyek alapvetően meghatározzák a nyílt forráskódú fejlesztések működését (Raymond 2007).

¹⁰ A szabad- és nyílt forráskódú szoftverfejlesztésben egy-egy szoftver forráskódjára alapozva külön fejlesztési elágazások (*fork*), új szoftvertermékek jöhetnek létre, melyek sok esetben a fejlesztői közösségen belüli eltérő fejlesztési elképzelések eredményeként jönnek létre. Az új elágazások létrejötte ezért többnyire valamilyen törést is jelképez a közösségen belül, melyre alkalmanként a „hitzakadás” (*schism*) kifejezés használatával is utalnak.

2. Szabad szoftver és nyílt forráskód mozgalom

A szabad szoftver elképzelése és a nyílt forráskód mozgalom kialakulása egymástól elválaszthatatlan. S bár a két kategória közötti különbség a hétköznapi használatban esetenként elmosódik, az elnevezésbeli különbségek komoly ideológiai és gazdasági háttérrel alapulnak. Feller és társai (2005) szerint szabad és a nyílt forráskódú szoftverrel kapcsolatos meghatározások alapvetően azonosak, a fogalomhasználattal kapcsolatos döntés az adott fejlesztői kultúrán belül pedig „általában inkább ideológiai, semmint funkcionális” (Feller et al. 2005: XVII). A szabad szoftver elképzelés hívei hajlamosak az erkölcsi súllyal is bíró szabadság (*freedom*) fogalmát hangsúlyozni, míg a nyílt forráskód mozgalom támogatói az angol *free* kifejezés ambivalenciája (szabad / ingyenes) miatt inkább a forráskód nyitott jellegére fókuszálnak.

A nyílt forráskódú szoftverfejlesztési modell az 1990-es évek második felében vált közismertté, és számos ponton alapoz a már korábban is működő modellre, a szabad-szoftver- fejlesztésre. Ahogyan azonban Himanen (2001) és Castells (2002) már korábbiakban kifejtett értelmezései is rámutatnak, az információs technológiák körében végbemenő változások a hálózati társadalom jóval szélesebb területén éreztetik hatásukat, jelentőségük nem korlátozható pusztán az informatikai ágazatokra.

A nyílt forráskódú fejlesztés hatása a gazdaságra az ezredforduló környékén kezdett érezhetővé válni. A kedvező médiareprezentációk, melyek jobbára a Microsoft Windows operációs rendszerének ingyenes konkurensét látták a nyílt forráskód éllavasaként számon tartott Linuxban, megfelelő lehetőséget biztosítottak ahhoz, hogy a modell addigi eredményei szélesebb körökben is ismertté váljanak.¹¹ Amellett, hogy a Red Hat Linux disztribúció 1999. augusztus 1-ji tőzsdére kerülése – mely eddig minden idők nyolcadik legnagyobb nyereségét hozta az első kereskedési napon – eleve ráirányította a figyelmet az újfajta fejlesztői modellre, több, nem feltétlenül az informatikához kapcsolódó területen is felfigyeltek a széles körű kollaborációban rejlő lehetőségekre. Tapscott és Anderson (2007) például leírja egy torontói aranybányászattal foglalkozó cég, a Goldcorp vezetésének szemléletváltását. A csőd szélén álló vállalat elnök-vezérigazgatója Rob McEwen Linus Torvalds, a Linux létrehozójának előadását hallva úgy döntött, nyilvánosságra hozza az interneten a cég korábban üzleti titokként kezelt geológiai vizsgálatainak eredményeit, és a jutalmat ígért annak, aki a legpontosabban meg tudja határozni, hogy hol, milyen technológiával lehetne gazdaságosan feltárni a feltételezett készleteket. McEwen a Linux operációs rendszerek kollektív intelligencia lefölöző hatását igyekezett így hasznosítani egy korábban zárt gazdasági területen.

A nyílt forráskódú fejlesztés közvetlen előzményeként mindenképpen a Richard M. Stallman által elindított GNU¹² projektet, valamint a szintén általa létrehozott Sza-

¹¹ A széles körű ismertség mindazonáltal nem járt együtt a nyílt forráskódú alkalmazások igazán széles körű elterjedésével – legalábbis a konkurensként kezelt Windowszal szemben.

¹² A GNU (a *Gnu is Not Unix* rekurzív mozaikszava) egy, a Unix alapvető struktúráját utánzó, de attól teljes mértékben függetlenül operációs rendszer lett volna, melyhez számos kiegészítő alkalmazást készítettek, de a rendszer stabil magja a mai napig nem készült el, ezért önálló operációs rendszerként nem működtethető. A GNU-hoz készült alkalmazások később a Linux alatt kerültek felhasználásra, melynél a rendszer magja (kernel) megbízhatóan működött, viszont nem voltak hozzá alkalmazások.

bad Szoftver Alapítványt (*Free Software Foundation*) jelölhetjük meg. Stallman 1971-ben kezdett dolgozni programozóként az MIT mesterségesintelligencia-kutató részlegén, és abban a korai hackerkultúrában szocializálódott, melyben a programok egymás közötti terjesztése, módosítása, javíthatása a programozói munka természetes részét képezte különösebb jogi konzekvenciák nélkül. A változást az 1980-as évek eleje hozta el, mikor az MIT lecserélte gépparkját, és a korábban használt DEC PDP-10 helyett VAX és Motorola 68020 számítógépeket kezdtek használni. Utóbbiak azonban zárt kódú, nem módosítható, nem terjeszthető operációs rendszerekkel működtek kivéve ezzel a felhasználók kezéből a továbbfejlesztés lehetőségét. Stallman (2002) ezen új érában egyenesen a korábbi programozói kultúra összeomlását látta.

Stallman 1984-ben ezért otthagyta korábbi munkahelyét és belekezdett egy saját operációs rendszer, a GNU fejlesztésébe, melyhez a projekt körül 1985-ben megszervezett Szabad Szoftver Alapítvány (*Free Software Foundation*) új jogi licencet dolgozott ki – ez lett a GPL (*GNU General Public Licence*). Stallman saját bevallása szerint döntését egy új operációs rendszer, és egy új szoftvertípus létrehozásához valamiféle belső erkölcsi kényszer szülte.

„Csinálhattam volna pénzt a zárt forráskódú szoftverek fejlesztéséből, és talán még el is szórakoztatott volna a kódolás. Tudtam azonban, hogy a karrierem végén visszatekintek majd azokra a falakra, amiket ezért építettem, hogy az embereket elválasszam egymástól, és úgy éreztem volna, hogy rosszabb helyre tettem a világot. (...) Olyan utat kerestem tehát, amin járva egy programozó a jó érdekében munkálkodhat” (Stallman 2002: 19).

Úgy érezte, hogy amennyiben elfogadja a zárt forráskód világát, „rosszabb ember” lesz, mert segít abban, hogy az embereket elszigetelje egymástól. Ehelyett a szabad fejlesztés útját választotta. Érdeemes megjegyezni, hogy a GPL licence alapvető elképzelése egy szójátékon¹³ alapul – Stallman copyright helyett sok esetben *copyleft*-ről beszél, melynek lényege nem az, hogy a szoftvert, mint alkotást magántulajdonként kezelje, hanem az, hogy elősegítse annak szabad terjedését és fejlődését.

A szabad szoftver filozófiájának lényege a programok felhasználásával kapcsolatos négy alapvető kitétel:

1. A szoftvert bármilyen célra szabadon lehet használni;
2. A szoftver a felhasználó saját igényei szerint szabadon módosítható (melyhez szükség van arra, hogy a felhasználó hozzáférhessen a szoftver forráskódjához);
3. A szoftver szabadon terjeszthető;
4. A szoftver módosított változatai szintén szabadon terjeszthetők, ezáltal a közösség is profitálhat a fejlesztésekből.

¹³ Stallman számos megnyilatkozásában visszaköszön a hackerkultúra játékosága, mely azonban saját radikális nézeteivel is keveredik. Stallman olykor saját „vallásának” az Emacs Egyházának (*Church of Emacs*) szentjeként, Szent IGNUciusként jelenik meg az általa tartott előadásokon – fején glóriával –, megáldja a hallgatóság számítógépeit, illetve a szabad szoftverek filozófiáját hirdeti. (Az Emacs egyébként a GNU-projekt egyik legfontosabb szövegszerkesztő alkalmazása.)

Stallman saját megfogalmazásában – mely nem egyezik meg például az alapítvány magyar oldalán megjelenített verzióval – ezen szabadságelvek a felhasználó szabadságán keresztül határozzák meg a program jellegét. A kitételek tehát nem közvetlenül a szoftverre vagy a tulajdonosra vonatkoznak, hanem arra, hogy a felhasználó *mit tehet* az adott alkotással (Stallman 2002: 20). Ez fontos különbséget jelent a nyílt forráskód definícióval szemben, mely inkább arra helyezi a hangsúlyt, hogy *minek kell megfelelnie* egy nyílt forráskódú szoftvernek.

A GPL és a szabad szoftver létrejötte nagyon fontos lépés volt a hackerkultúra életében, hiszen újabb lehetőségeket nyitott meg a fejlesztés terén. A szabad szoftver jó gyakorlati példája a szabadság iránti vágy kinyilvánításának, illetve a hackerek munka- és pénzetikájának. Stallman felfogásában a szabad szoftver elsősorban társadalmi szoftver, amennyiben hozzásegíti a szélesebb társadalmi rétegeket az informatikai eszközök felhasználásához, alkotói szabadságuk gyakorlásához. S bár Stallman folyamatosan hangsúlyozza, hogy szabad szoftver lényege nem az ingyenesség (ez csak a szabad felhasználás hozadéka), a társadalmi recepció gyakorta elsősorban erre a tulajdonságra összpontosul. Ez pedig üzleti szempontból nem kedvez azon fejlesztőknek, akik mégis szeretnék tőkévé konvertálni tudásukat.

A szabad szoftverek ingyenessége nem jelenti azt, hogy ne lehetne belőlük profitot termelni, ám a profitrealizálás itt nem a termékben, hanem a kapcsolódó szolgáltatásokban (átalakítás, javítás, telepítés) van. A szabad szoftver fent említett ingyenessége mégis olyan, a for-profit szféra szempontjából negatív konnotációkat hordoz, mint az „ingyenes” és a „gyenge minőségű” jelzők. Raymond (2004) emellett arra is felhívja a figyelmet, hogy Stallman és a Szabad Szoftver Alapítvány aktivistái olyan radikális kereskedelmiszoftver-ellenes véleményt fejtenek ki, ami szükségessé tette egy új, gazdaságilag semlegesebb koncepció bevezetését. Vainio és Vadén (2007) szerint Stallman szabad szoftverekről alkotott elképzelései, radikális kritikája a hagyományos birtokviszonyokkal Thomas Hobbes, John Locke vagy John Stuart Mill liberális politikai filozófiájára épül, és túllép a szoftverfejlesztés határain. Stallman afféle politikai állásfoglalásként, programként fogalmazza meg saját elképzeléseit a programozásról, amely az egyik legfontosabb törésvonalat jelenti a szabad, és a nyílt forráskódú fejlesztés között.

Bár az 1991-ben elindult Linux szintén a szabad szoftveres GPL-licenct használja, az operációs rendszer ma a nyílt forráskódú fejlesztés legkiemelkedőbb példaként említhető. A nyílt forráskódú fejlesztés strukturálisan jobbra megegyezik a szabad szoftver fejlesztéssel: nyitott, önszervező projektszemléleten, kollaboráción alapul, és hasonló lehetőségeket biztosít az alkotások szabad változtatására és terjesztésére. A legfontosabb különbség a szabad és a nyílt forráskódú fejlesztés között a szoftverkészítés gazdaságpolitikai háttere. A szabad szoftver ideológiai, politikai terheltségével szemben a nyílt forráskód ideológiailag semlegesebb, „üzletbarát” fejlesztési modellként hivatkozik önmagára, mely minőségi alternatívát igyekszik nyújtani a kereskedelmi szoftverekkel szemben (Vainio & Vadén 2007). A Bruce Perens (1999) által megfogalmazott nyílt forráskód definíció (open source definition) – mely eredetileg a Debian Linux disztribúció saját jogi licenceként jelent meg – tíz fontosabb jellemzőt emel ki, melyek a *szoftverre és a szoftverhez fűződő licencre* vonatkozó elvárások. Itt a hangsúly tehát nem a felhasználó vagy a fejlesztő lehetőségeinek szabályozásán, hanem a programmal, illetve a licence-szel kapcsolatos elvárásokon van. A nyílt forráskód definíció sokkal inkább

technikai jellegű szöveg, mely olyan fontosabb többlettartalmakkal bír, mint például a technológiasemlegesség (platformfüggetlenség), vagy a más alkalmazásokra kiterjedő megkötések tiltása. Utóbbi azért is fontos, mert ez a kitétel jelzi a nyílt forráskód kezdeményezés azon törekvését, hogy a kereskedelmi szoftverekkel együtt is használható legyen.

A szabad szoftver és a nyílt forráskód közötti különbséget jól szemlélteti Yeats (2007), aki Stallman és Raymond megnyilatkozásait állítja szembe egymással:

„Míg Stallman esetében a szabad szoftver népszerűsítésében a hangsúly nyilvánvalóan az erkölcsös viselkedésen és a kötelességen van, Raymond a nyílt forráskód bemutatásakor a decentralizált fejlesztési folyamat technológiai felsőbbrendűségét helyezi a középpontba. Stallman úgy festi le a szabad szoftvert, mint ami felsőbbrendű és etikai szempontból megfelelőbb alternatíva a kereskedelmi programfejlesztésnél, amely a felhasználó, illetve a fejlesztő jogaira és szabadságára fókuszál. Raymond írásaiban sehol nem állítja, hogy a kereskedelmi szoftver etikátlanabb választás lenne. Persze nem mondhatjuk, hogy Raymond ne fogalmazna meg kritikát a kereskedelmi szoftverekkel szemben, számára azonban a szoftver technológiai implikációi mindig is fontosabbak a morális implikációjánál” (Yeats 2007: 29–30).

Raymond (2004) ezen szembenállást a hackeretika radikális és a pragmatikus értelmezése közti különbségnek nevezi, mely szerinte szükségessé tette, hogy a hackerkultúra tudásának gazdasági haszonná konvertálhatósága érdekében a nyílt forráskód már elnevezésében is elhatárolódjon a stallmani radikalizmustól.

3. Az Ubuntu Linux

Az 1990-es évek változásai, melyek szükségessé tették a szabad és nyílt forráskódú fejlesztés elválását – ha másként nem is, retorika szintjén mindenképpen – valójában pusztán előrejelzői voltak azoknak a változásoknak, amelyeket az ezredfordulót követően tapasztalhattunk. A nyílt forráskódú fejlesztés, és a kapcsolódó szolgáltatások ugyanis nemcsak sikeres üzletté váltak, hanem részei lettek a mindennapi életnek. A Linux alapú operációs rendszerek például már a szakértő felhasználók köreit jelentősen meghaladó körökben is használatban vannak. Jó példája a tömeges használatra kialakított Linux változatoknak az Android mobil operációs rendszer is, az alábbiakban ugyanakkor egy másik kortárs nyílt forráskódú projektre térnek ki részletesebben – az Ubuntu Linuxra. Utóbbi azért különösen érdekes példa, mert jól szemlélteti a készítőik azon törekvéseit, hogy a hagyományos hackerkultúra értékeit hangsúlyozva mégis szembeszálljanak a korábbi moralizáló, elitista felfogással, és egy piacképes terméket hozzanak létre. Ez a törekvés azonban láthatóan feszültségekhez vezet az Ubuntu fejlesztői, használói, illetve más Linux projektek tagjai között. Az Ubuntu példája azért is különösen érdekes, mert az operációs rendszerrel szemben megfogalmazott kritikák gyakran ugyanazokat az etikai alapelveket és fejlesztési hagyományokat kérik számon a Canonicalon, amelyeket a cég használ fel saját imázsának fenntartására és döntéseinek alátámasztására.

Az Ubuntu Linux disztribúció megjelenése és az eltelt tíz év tapasztalatai jó példával szolgálnak arra a nézetváltásra, mely a nyílt forráskód mozgalom megjelenésével elkezdődött a szoftverfejlesztésben. A nyílt forráskódú operációs rendszer jól reagál azokra a társadalmi változásokra, melyek az infokommunikációs technológiákkal kapcsolatban érzékelhetők az elmúlt évtizedben, és egy, a korábbiaktól eltérő szemléletet igyekszik meghonosítani a szabad és nyílt forráskódú fejlesztés világában. Emellett azonban az operációs rendszer relatív népszerűsége, a felhasználók és a fejlesztők közötti feszültségek is kérdéssé teszik a hackeretika mint ideológiai keretrendszer életképességét.

Az Ubuntu Linux fejlesztésének hátterét a Mark Shuttleworth által alapított és támogatott Canonical Ltd. biztosítja. A céget Shuttleworth azzal az alapvető céllal hozta létre, hogy pénzügyi és technikai támogatást nyújtson az Ubuntu fejlesztéséhez, illetve bevételt szerezzen az operációs rendszerhez kapcsolódó szolgáltatások által. Emellett, a többi Linux-disztribúcióhoz hasonlóan az Ubuntu fejlesztésében is komoly szerepe van a független fejlesztői közösségnek, az önkénteseknek, akik fordításokkal, hibajelzésekkel, javításokkal, vagy alkalmazások tervezésével, fejlesztésével járulnak hozzá a közös alkotáshoz.

Az Ubuntu elnevezés egy bantu kifejezésből származik, melynek durva fordítása „emberségesség másokkal szemben”. Az eredeti kifejezéshez kapcsolódó értelmezés a mai napig fontos részét képezi a projekt önmeghatározásának és a brand retorikájának. Az operációs rendszer fejlesztői szinte minden platformon igyekeznek hangsúlyozni, hogy az általuk fejlesztett rendszer a közösség, a felhasználók érdekeit tartja szem előtt.

Az Ubuntu az utóbbi tíz év során az egyik legismertebb Linux disztribúcióvá nőtt ki magát. A Distrowatch.com által nyilvántartott 262 aktív Linux-projekt közül a második legnépszerűbb operációs rendszer, melyre 71 további disztribúció alapul.¹⁴ A 2004 októberében útjára indított operációs rendszer kezdetben a Debian nevű disztribúcióra, illetve a Gnome grafikus felületre épült, fejlődése azonban mára saját irányt vett. Az Ubuntu hátterében álló elképzelés nem pusztán egy, a kereskedelmi, zárt forráskódú operációs rendszerek melletti alternatíva, hanem egy versenyképes termék létrehozása. Mark Shuttleworth maga is hangsúlyozta, hogy a Linux operációs rendszerek egyik leggyengébb pontja az, hogy a műszaki tervezésbe, programozásba fektetett energia és szakértelem nem tükröződik a szoftverek megjelenésében. Ahogyan a hivatalos Ubuntu kézikönyvhöz írott előszavában fogalmaz, a szélesebb értelemben vett cél az, hogy a szabad szoftverek világa ne csak a megbízhatóságról, a teljesítményről és a biztonságról szóljon – „ideje a keverékhez adni a könnyű használhatóságot és a stílusosságot is” (Hemke & Graner 2012: xix).

Egy 2014-es interjúban Mark Shuttleworth úgy nyilatkozott, hogy bár az Ubuntu felhasználóinak száma komolyan megnőtt, az operációs rendszer maga nem vált a mainstream részévé. Meglátása szerint nyugaton inkább az IT-szektorban tevékenykedő szakemberek, és a „jól informáltak [...], a szerencsés kevesek” (Oxford 2014: 42) körében használják, míg keleten jóval szélesebb körben elterjedt a végfelhasználók szintjein. Mivel a keleti piacok kapcsán Shuttleworth Indiát, Pakisztánt és Kínát emeli

¹⁴ 2014. július 10-i adatok alapján.

ki, hozzátehető, hogy utóbbiak esetében az Ubuntu népszerűségét minden bizonnyal anyagi, illetve jogi kérdések is befolyásolják.

Az Ubuntu fejlesztésében kétfajta vízió keveredik: a nyílt forráskód mozgalomban egyaránt jelen lévő közösségi, illetve üzleti szemléletű elképzelések. Bár a Linux operációs rendszerek fejlesztésében nem újdonság, hogy egy nagyobb informatikai cég támogatja a közösségi projekteket, az Ubuntu esetében sajátosnak mondható helyzet alakult ki. A fejlesztések hátterében álló cég, a Canonical ugyanis sok esetben szembehelyezkedik a független fejlesztők és felhasználók közösségének elvárásaival a fejlesztés irányának tekintetében. Az utóbbi években a Canonical döntései jócskán megosztották az operációs rendszer körül kialakult közösséget, és számos kérdést felvetettek a nyílt forráskódú fejlesztések jövőbeli irányával és lehetőségeivel kapcsolatban. Shuttleworth személyét is gyakran éri támadás az üzleti döntések miatt, ami nem véletlen, hiszen az alapító a mai napig kezében tartja az Ubuntu fejlesztésének irányítását.

Az Ubuntu fejlesztése terén a Canonical egy sajátos döntést hozott: az operációs rendszert ugyan egy professzionális csoport készíti, ugyanakkor – a Red Hat vagy a Novell által támogatott SuSE Linux példájától eltérően – nem bontják külön ingyenes közösségi, illetve céges változatokra. A felhasználói oldalról mindez előnyösnek is tűnhet, hiszen nincsenek olyan extra funkciók, amiért az átlagos felhasználónak fizetnie kellene, az elképzelés azonban számos ponton szembemegy a hackerkultúra ideológiai örökségével, ami feszültséget generál a felhasználók közösségén belül.

Jane Silber, a Canonical jelenlegi vezérigazgatója 2012-ben így nyilatkozott a struktúra kialakításáról:

„[...] úgy gondoltuk, hogy üzleti modellünknek nem a szoftver részeire, hanem a köré épített szolgáltatásokra kell épülnie. Így korán elköteleztük magunkat egy olyan döntés mellett, hogy a legjobb munkánk mindig szabadon hozzáférhető lesz, és nem lesz kevésbé jó verzió a közösség, illetve jobb változat az üzleti, vagy fizető felhasználók számára. A biztonsági frissítések és a szoftver további összetevői helyett mi a szolgáltatásainkat áruljuk” (Ubuntu 2012: Unity, Community and 'Continuous Computing' 2012: 19).

Silber maga is rámutatott arra a tudatos üzleti döntésre, mely a projektet megkülönbözteti a Red Hat vagy a Novell fejlesztési elképzeléseitől. Ennek a nagylelkűségnek ugyanakkor ára van: a Canonical ugyanis a jelek szerint egyre kevesebb beleszólást enged a fejlesztésekbe a közösség oldaláról. Silber maga is beismeri, hogy a felhasználói közösség és a cég, valamint a részvényesek érdekei közötti különbség feszültséget eredményez, amelyet ugyanakkor „egészséges”, hasznos feszültségként értelmez (Ubuntu 2012: Unity, Community and 'Continuous Computing' 2012).

Susan Linton (2013) a *Linux Format* magazinban megjelent cikkében igyekszik összefoglalni azokat a problémákat, amelyek miatt véleménye szerint az Ubuntu folyamatosan elveszti a kapcsolatot saját felhasználói közösségével. Az operációs rendszerrel kapcsolatos problémák elsődleges forrása láthatóan az a különbség, mely a közösség és a Canonical között fennáll. Bár a Canonical retorikájában a kezdetektől fontos elem a közösségi jelleg kihangsúlyozása, az utóbbi évek fejlesztései sok esetben figyelmen kívül hagyták a közösségi érdekeket. A grafikus kezelőfelület leváltása, a kollaborációt segítő

portál forráskódjának zártan tartása, az Ubuntu TV és mobil operációs rendszer felé való elmozdulás nemcsak technikai, hanem etikai oldalról is problémákat jelentettek, legálábbis az Ubuntu korai felhasználói számára.

Az egyik vád az Ubuntuval kapcsolatban az, hogy a Canonical inkább elvesz a közösségtől, semmint visszaad. Amint azt korábban láthattuk, a hackerkultúrában komoly szerepe van a reciprocitásnak, illetve a közösségnek adott ajándéknak. A nagyobb piaci szereplők „ajándéka” a nyílt forráskódú projektek esetében részben az a munka, amivel maguk is hozzájárulnak a közös Linux rendszermag (kernel) fejlesztéséhez, hibajavításához. Linton (2013) felidézi, hogy a Canonical 2008-ban mindössze a javítások 0,1 százalékaival járult hozzá a közös fejlődéshez, szemben a Red Hat (a Red Hat, illetve a Fedora disztribúciók fejlesztője) 11,9, illetve a Novell (a SUSE Linux fejlesztője) 7,3 százalékos arányával. Ezzel pedig épp az ajándékkultúra altruista jellegét, a közösségért vállalt felelősséget, a bizalom alapjait kockáztatják. A Canonical elsősorban saját rendszerének fejlesztésére koncentrál, és a kritikák szerint jóval több hasznot húz mások munkájából, mint amennyit maga hozzátesz a Linux közös fejlesztéséhez.

Az a moralista kritika viszont, amely a kölcsönösség hiányát, a kiegyensúlyozatlanságot kéri számon az operációs rendszer fejlesztőin, sok esetben azon Linux felhasználóktól érkezik, akik már tapasztalt, hozzáértő felhasználóként tekintenek az utóbbi évek fejleményeire, és valamiféle elitista szemléletet szívtak magukba. A hackerkultúra örökségének védelmezése kerül szembe ez esetben a Canonical haszonelvűségével. Aho-ghan Shuttleworth (lásd: Oxford 2014) és Silber (lásd Ubuntu 2012: Unity, Community and 'Continuous Computing' 2012) a képet kiegészítve hangsúlyozni próbálja, a haszonelvűség a felhasználó közösség érdekét szolgálja, amennyiben az továbbra is szabadon és ingyen juthat hozzá a jó minőségű, felhasználóbarát operációs rendszerhez.

Richard Stallman, a Szabad Szoftver Alapítvány létrehozója egy 2013-as interjúban az Ubuntuval, és számos más Linux disztribúcióval kapcsolatban kihangsúlyozta, hogy ezen operációs rendszerek sok esetben fontos etikai normákat sértenek meg azzal, hogy nemcsak szabad, de zárt szoftvereket is tartalmaznak. Ezzel Stallman szerint azt bizonyítják, hogy számukra nem a felhasználó szabadsága, hanem a kényelme az elsődleges. Meglátása szerint így azt tanítják a felhasználóknak, hogy a szabadságot kevesebbre értékeljék, mint a kényelmet. Ráadásul az Ubuntu legújabb fejlesztései a kémprogramokhoz hasonló tulajdonságokat kölcsönöznek a rendszernek („Richard Stallman Talks About Linux” 2013). Az operációs rendszer 12.10-es verziójában megjelent általános keresés kiszivárogtatja a felhasználó keresési kulcsszavait és IP-címét. Az általános kereső ugyanis nem pusztán a felhasználó eszközén található adatok, hanem online áruházak (például az Amazon) kínálatában, közösségi alkalmazásokon (például Facebook, Twitter), illetve egyéb szolgáltatásokon (például a BBC oldala) elérhető tartalmak között is keres, és ezek közül is ajánl találatokat (Micah 2012).

A Stallman és Micah által megfogalmazott problémák szintén a hackeretika moralista felfogásán alapulnak. Nem csupán azt kérik számon a Canonicalon, hogy a cég a felhasználók érdekei helyett saját üzleti megfontolásait helyezi előtérbe a fejlesztések során, hanem egyenesen azt, hogy az Ubuntu a zárt forrású, kereskedelmi szoftverek mintáját követi, és kényelmes fogyasztóvá alacsonyítja a felhasználót. A stallmani kritika tehát végső soron nem a technikai gyakorlatra, hanem a kényelem csalfa ígéretére, a fogyasztói társadalomra vonatkozik.

A fenti kritikákkal szemben ugyanakkor felhozható, hogy az Ubuntu fejlesztői szembemennek a tradícióval, nem félnek kiszakítani a Linuxot a szakértő felhasználók szűk köréből. Everard (2013) szerint a Canonical pozitív példával jár elől, amikor arra törekszik, hogy a Linuxot versenyképes, jól kezelhető, átgondolt és kidolgozott megjelenítésű, felhasználóbarát operációs rendszerré tegye, és ezzel új „neutrális” szereplőként jelenjen meg a Microsoft, az Apple, vagy a Google mellett. Ezen szerepet látszik erősíteni az a törekvés is, hogy az Ubuntu az asztali operációs rendszer helyett afféle általános, többfajta eszközön működtethető operációs rendszerré kíván fejlődni. A Canonical – és személy szerint Mark Shuttleworth – kommunikációjának gyakori visszatérő eleme az innovatív jelleg, valamint a felhasználóbarát, egyszerű felület hangsúlyozása. Ezen jellemzők kiemelése pedig egyértelműen azt jelzi, hogy a Canonical érzékeli az infokommunikációs eszközök társadalmi elterjedtségével, a felhasználó praxisokkal kapcsolatos változásokat.

Az Ubuntuval kapcsolatos ilyen jellegű megközelítés végső soron arra világít rá, hogy a hackerkultúrából örökölt etikai alapok érdemben átléphetők, ha azok az adott terület fejlődését hátráltatják. A terület fejlődésének pontos értelmezésére alapvetően eltérő megközelítésekkel találkozunk a szabad- és nyílt forráskódú fejlesztésben. A tradicionálisabb fejlesztői közösségek a hackerkultúra elitizmusával egybecsengően inkább az operációs rendszer technológiai fejlődését tekintik célnak, míg a Canonical víziójában a versenyképes, felhasználóbarát, gazdasági profitot biztosító termék kifejlesztése a lényeg. Ez pedig egy érdekes, de nem feltétlenül új ellentmondást jelent a nyílt forráskódú fejlesztésben: a közösség érdekeit a nyitottság, a transzparencia kapcsán szembe kerülhetnek a közösség érdekeivel a technológia fejlődésével kapcsolatban.

Összegzés

A számítástechnika kialakulásának társadalmi környezetét biztosító akadémiai közeg ideái a korai hacker kultúrán át a mai szabad és nyílt forráskódú szoftverfejlesztő projektekig megmaradtak. Az információszabadsággal, a tudás, az eredmények szabad megosztásával, a közösség fejlődésének előremozdításával kapcsolatos elképzelések, elvárások máig biztosítják a fejlesztések kereteit, és komoly viták, feszültségek forrásaivá is válhatnak. Az Ubuntu Linux jó példája annak, hogy a kortárs nyílt forráskódú fejlesztésben legalább a retorika szintjén megmaradtak azok az altruista, közösségi elemek, melyek korábban az akadémiai körökben a tudományos innovációk szempontjából instrumentálisak voltak. A Linux operációs rendszerek további fejlődése szempontjából mindenképpen érdemes számot vetni azzal, hogy az Ubuntuhoz hasonló kezdeményezések egyre inkább feladják a korábbi moralista, közösségközpontú szemléletet, és inkább a gazdasági versenyképességet teszik elsődleges céljá.

Irodalom

- Bergquist, Magnus & Ljungberg, Jan (2001) The power of gifts: organizing social relationships in open source communities. *Info Systems* 11. 305–320. o.
- Castells, Manuel (2002) *Az Internet-galaxis*. Budapest, Network TwentyOne.
- Castells, Manuel (2005) *A hálózati társadalom kialakulása*. Budapest, Gondolat – Infonia.
- Castells, Manuel (2006) *Az identitás hatalma*. Budapest, Gondolat – Infonia.
- Everard, Ben (2013) Why Ubuntu's great. *Linux Format July 2013*. 48–49. o.
- Feller, Joseph et al. (2005) szerk. *Perspectives on Free and Open Source Software*. Cambridge – London, MIT Press.
- Flammich Mariann (2002) Hackerek. *Médiakutató 2002 ősz*. (online) Elérhető: http://mediakutato.hu/cikk/2002_03_osz/01_hackerek Hozzáférés: 2011. december 15.
- Hemke, Matthew & Graner, Amber (2012) *The Official Ubuntu Book*. Boston – Indianapolis – San Francisco, Prentice Hall.
- Himanen, Pekka (2001) *The Hacker Ethic and the Spirit of the Information Age*. New York – Toronto, Random House.
- Kazári Csaba (2003) *Hacker, cracker, warez*. Budapest, Computer Panoráma.
- Levy, Steven (2010) (1984). *Hackers – Heroes of the Computer Revolution*. Sebastopol, CA, O'Reilly Books.
- Lin, Yu-Wei (2007) Hacker Culture and the FLOSS Innovation. In St. Amant, Kirk & Still, Brian (szerk.). *Handbook on Research on Open Source Software: Technological, Economic, and Social Perspectives* 34–46. o. Hershey – New York, Information Science Reference.
- Linton, Susan (2013) Has Ubuntu lost it? *Linux Format July 2013*. 42–47. o.
- Micah, Lee (2012) Privacy in Ubuntu 12.10: Amazon Ads and Data Leaks. (online) *Electronic Frontier Foundation*. Elérhető: <https://www.eff.org/deeplinks/2012/10/privacy-ubuntu-1210-amazon-ads-and-data-leaks> Hozzáférés: 2014. június 21.
- Nissenbaum, Helen (2004) Hackers and the contested ontology of cyberspace. *New Media & Society* Vol.6 (2.) 195–217. o.
- Oxford, Tamsin (2014) The world according to Mark. *Linux Format April 2014*. 42–45. o.
- Perens, Bruce (1999) The Open Source Definition. (online) In Chris DiBona et al. (szerk.) *Open Sources: Voices from the Open Source Revolution*. Elérhető: <http://oreilly.com/catalog/opensources/book/perens.html> Hozzáférés: 2014. július 29.
- Raymond, Eric S. (2004) *A katedrális és a bazár*. Budapest, Kiskapu.
- Richard Stallman Talks About Linux. (online) Elérhető: <http://www.youtube.com/watch?v=CP8CNp-vksc> Hozzáférés: 2014. június 21.
- Smajda, Jon (2011) Open Source and the Moral Field of Computing. *Journal of Information Technology & Politics* 8:3. 304–322. o.
- Stallman, Richard M. (2002) *Free Software, Free Society: Selected Essays of Richard Stallman*. Boston, GNU Press – Free Software Foundation.
- Stewart, Katherine J. & Gosain, Sanjay (2006) The Impact of Ideology on Effectiveness in Open Source Software Development Teams. *MIS Quarterly* Vol. 30 No. 2. 291–314. o.
- Tapscott, Don & Williams, Anthony D. (2007) *Wikinómia*. Budapest, HVG Könyvek.
- Taylor, Paul A. (1999) *Hackers – Crime in the digital sublime*. London – New York, Routledge.
- Thomas, Douglas (2004) *Hacker Culture*. Minneapolis – London, University of Minneapolis Press.
- Turkle, Sherry (2005) *The Second Self – Computers and the Human Spirit*. Cambridge, Mass. – London, The MIT Press.
- Ubuntu (2012) Unity, Community and 'Continuous Computing'. *Linux User & Developer* 112. 16–21. o.

- Ubuntu (2012) Unity, Community and 'Continuous Computing'. *Linux User & Developer* 112. 16–21. o.
- Vainio, Niklas & Vadén, Tere (2007) Free Software Philosophy and Open Source. In St. Amant, Kirk & Still, Brian (szerk.). *Handbook on Research on Open Source Software: Technological, Economic, and Social Perspectives* 1–11. o. Hershey – New York, Information Science Reference.
- Yeats, David (2007) Morality and Pragmatism in Free Software and Open Source. In St. Amant, Kirk & Still, Brian (szerk.). *Handbook on Research on Open Source Software: Technological, Economic, and Social Perspectives* 23–33. Hershey – New York, Information Science Reference.