

Kratka povijest UNIXa Od UNICSa do FreeBSDa i Linuxa

Autor: Hrvoje Horvat

Naslov: Kratka povijest UNIXa - Od UNICSa do FreeBSD i Linuxa

Licenca i prava korištenja: Svi imaju pravo koristiti, mijenjati, kopirati i štampati (printati) knjigu, prema pravilima [GNU GPL](#) licence.

Mjesto i godina izdavanja: Osijek, 2017

ISBN: 978-953-59438-0-8 (PDF-online)

URL publikacije (PDF): [https://www.opensource-osijek.org/knjige/Kratka povijest UNIXa - Od UNICSa do FreeBSD i Linuxa.pdf](https://www.opensource-osijek.org/knjige/Kratka%20povijest%20UNIXa%20-%20Od%20UNICSa%20do%20FreeBSDa%20i%20Linuxa.pdf)

ISBN: 978-953- 59438-1- 5 (HTML-online)

DokuWiki URL (HTML): <https://www.opensource-osijek.org/dokuwiki/wiki:knjige:kratka-povijest-unixa>

Verzija publikacije : 1.0

Naklada : Vlastita naklada

→ Uz pravo svakoga na vlastito štampanje (printanje), prema pravilima GNU GPL licence.

Ova knjiga je napisana unutar inicijative **Open Source Osijek**: <https://www.opensource-osijek.org>

Inicijativa **Open Source Osijek** je član udruge **Osijek Software City**: <http://softwarecity.hr/>

UNIX je registrirano i zaštićeno ime od strane tvrtke **X/Open (Open Group)**. **FreeBSD** i **FreeBSD** logo su registrirani i zaštićeni od strane **FreeBSD Foundation**. Imena i logo : **Apple, Mac, Macintosh, iOS i Mac OS** su registrirani i zaštićeni od strane tvrtke **Apple Computer**. Ime i logo **IBM i AIX** su registrirani i zaštićeni od strane tvrtke **International Business Machines Corporation**. **IEEE, POSIX i 802** registrirani i zaštićeni od strane instituta **Institute of Electrical and Electronics Engineers**. Ime **Linux** je registrirano i zaštićeno od strane Linusa Torvaldsa u Sjedinjenim Američkim Državama. Ime i logo : **Sun, Sun Microsystems, SunOS, Solaris i Java** su registrirani i zaštićeni od strane tvrtke **Sun Microsystems**, sada u vlasništvu tvrtke **Oracle**. Ime i logo **Oracle** su u vlasništvu tvrtke **Oracle**. Ime i logo : **XinuOS, OpenServer, SCO UNIX, UnixWare i Open Desktop** su registrirani i zaštićeni od strane tvrtke **XinuOS**. Ime i logo **HP-UX** je registrirano i zaštićeno od strane tvrtke **Hewlett Packard Enterprise**. Ime i logo : **Cisco Systems, Cisco, IOS-XR i NX-OS** su registrirani i zaštićeni od strane tvrtke **Cisco Systems**. Ime i logo : **Dell, Dell Networking, DNOS i FTOS** su registrirani i zaštićeni od strane tvrtke **Dell**. Ime i logo : **Intel i Hyper-Threading** su registrirani i zaštićeni od strane tvrtke **Intel**. Ime i logo : **AMD, Zen i Ryzen** su registrirani i zaštićeni od strane tvrtke **AMD**.

Sadržaj

Predgovor	4
O Autoru	5
Povijest razvoja UNIXa	6
Vratimo se u povijest razvoja Unixa	8
1969 godina	9
1970 godina	10
Programski jezik C	11
Podrijetlo vrsta	13
AT&T UNIX System III i UNIX System V	15
Berkeley Software Distributions (BSD)	19
Priča o tvrtki Sun Microsystems	28
Druge paralelne priče	32
GNU i Linux	32
Od početka do danas	36
Unix Ratovi	37
Rezimirana povijest Unix-a	38
Izvori informacija	40
Više informacija o knjizi	41

Predgovor

U daljem tekstu upoznati ćemo se s povijesti razvoja operativnog sustava Unix, kao i njegovih raznih varijanti. Govorit ćemo i o pojedincima i grupama ljudi koji su omogućili nastanak Unixa te raznih tehnologija, koje su bile nužne i za njegov nastanak ali i dalji razvoj. Iste tehnologije koje su danas u upotrebi i u svim drugim operativnim sustavima. Dodatno, spomenuti ćemo i sveučilišta, koja su također dala veliki doprinos u razvoju Unixa, direktno ili indirektno.

Naglasak će biti na Unixu i njegovim raznim varijantama, uz vrlo malo spominjanje Linuxa iako je i on Unixoidni operativni sustav. Dobro, nećemo moći zaobići dio priče o Linuxu u potpunosti, ali se na njemu nećemo puno zadržavati.

Naime u posljednje vrijeme se puno govori o Linuxu a pomalo zaboravlja povijest Unixa i njegovih direktnih nasljednika, koji su na kraju krajeva i doveli do razvoja Linuxa.

U nekim slučajevima, pratit ćemo i više priča paralelno, jer su se ili razvijale u isto vrijeme ili su imale iste ciljeve. Spomenuti ćemo i sva dostignuća navedenih pojedinaca.

Ova knjiga bi trebala biti zanimljiva kako korisnicima *Windows* ili *Mac* računala, tako i korisnicima neke od distribucija *Linuxa* ili nekog drugog *Unixa*. Također ju preporučujem i svima ostalima, koji se koriste IT tehnologijama - možda i nešto naučite ili proširite vidike 😊

Pošto je ova knjiga objavljena prema GNU GPL licenci, imate ju pravo kopirati, mijenjati i tiskati (štampati), te vas ovim putem i potičem na to. Pozivam Vas da i sami date doprinos dijeljenju znanja. Izdvojite samo djelić svoga vremena i znanja te ga podijelite s drugima

“ Budi promjena koju želiš vidjeti u svijetu !. ”

Mohandas Karamchand Gandhi (Mahatma Gandhi)

O Autoru



Hrvoje Horvat je rođen 1975 godine u Osijeku, gdje je završio Prvu tehničku školu, smjer elektronika i automatika. Od prvog druženja s računalom, sredinom 80-tih ([ZX Spectrum](#)) i prvih igara a potom prelaskom na [Atari ST](#) i kasnije na PC arhitekturu, odabir budućeg zanimanja mu je postao vrlo jasan. Prvo umrežavanje računala s kolegama, u naselju, tijekom 1994, doprinijelo je novim smjerovima osobnog rada i razvoja. Uslijedili su razni poslovi održavanja računala i mreža, te se 2000. godine zapošljava u tvrtki **Siemens d.o.o** u kojoj i danas radi.

Prvih nekoliko godina radi kao sistem administrator, potom sistem i mrežni inženjer, na korporativnoj IT infrastrukturi tvrtke, te na nekoliko većih infrastrukturnih projekata. Pohađa razne specijalizacije, kako vanjske tako i unutar tvrtke. Zadnjih desetak godina radi kao razvojni inženjer, na projektima koji obuhvaćaju : visoko dostupne i redundantne sustave, AAA i PCRF servise, Virtualizaciju i NAS/SAN sustave te napredne mrežne servise i protokole.

Suosnivač je udruge za razvoj IT sustava *Udruga01*, unutar koje se pokreću deseci projekata razvoja i testiranja raznih mrežnih uređaja, protokola i tehnologija, u cilju učenja i stjecanja novih znanja. Suosnivač je inicijative „Open Source Osijek“, koja potiče i promovira razvoj i upotrebu sustava otvorenog koda, u kojoj i aktivno objavljuje stručne članke. Član je udruge *Osijek Software City* u kojoj djeluje kao aktivni član i predavač.

Povijest razvoja UNIXa

Priča o nastanku UNIXa prati nevjerojatne ljude, vizionare i vrhunske inženjere. Ljude koji su bili spremni odreći se puno toga, na putu k ostvarenju svojih vizija. Nevjerojatne pojedince i grupe ljudi koji su cijelo to vrijeme bili puni entuzijazma te nisu bili spremni na kompromise. Upoznati ćemo se (ili se podsjetiti) ovih stvarnih velikana IT svijeta koji su promijenili naš svijet a da toga možda nismo niti svjesni.

Bez njih ne bi bilo niti interneta (barem onakvog kakav danas poznajemo), Unixa, Linuxa, Mac OS-a, ili iPhone-a i Androida a ne bi bilo niti BlackBerry-ja. Ne bi bilo niti programskog jezika **C**, kao niti jezika koji su nastali na osnovi programskog jezika C:

- **C++**
- **D**
- **Go**
- **Rust**
- **Java**
- **JavaScript**
- **Limbo**
- **LPC**
- **C#**
- **Objective-C**
- **Perl**
- **PHP**
- **Python**
- **Swift**
- **Verilog**
- i mnogih drugih.



Samo programski jezik Java danas koristi preko 3 milijarde računala i raznih uređaja : *od set-top box uređaja, preko pisača, web kamera, navigacijskih sustava, raznih terminala, medicinskih uređaja, kućanskih aparata, televizora i sl.*

Dodatno Java "Virtual Machine" koriste i drugi programski jezici, poput jezika:

- **Scala**
- **Groovy**
- **JRuby** - implementacija Ruby jezika
- **Jython** - implementacija Python jezika

*I programi za Android se pišu u programskom jeziku Java, uz korištenje Android biblioteka. Potom se kod prevodi (Engl. compile) u bytecode za **Java Virtual Machine**, koji se zatim prebacuje u bytecode za Android Runtime (ART) iz kojega se i pokreću svi Android programi.*

Java Virtual Machine (JVM) odnosno Java virtualni stroj je program koji interpretira i prevodi Java program u Java bytekod, a koji se potom može izvršiti (pokrenuti).

Osim toga ne bi bilo niti operativnih sustava koji su danas u upotrebi (ili bi bili znatno lošiji), te raznih, vrlo važnih tehnologija koje se nalaze ugrađene u sve današnje operativne sustave.

Dodatno i većina operativnih sustava je razvijena u **C** jeziku ili nekoj njegovoj "izvedenici", a ne zaboravimo i na upravljačke programe (Engl. Driveri) za iste sustave, koji se također moraju razvijati u nekom programskom jeziku poput C, C++ ili sl.



Možda niste znali ali [Apple](#)ovi **MacOS** i **iOS** su također bazirani na određenim varijantama UNIXa (pogledajte poglavlje [Drugi nasljednici BSDa](#) - na kraju knjige)

Ne zaboravimo niti sve mrežne uređaje koji su okosnica Interneta, od Routera (usmjerivača), switcheva (preklopnika) , Firewalla (vatrozida), IPS i IDS uređaja i sl. Svi oni upogonjeni su u vrlo velikoj mjeri nekom varijantom UNIXa ili Linuxa.

Tako primjerice [Cisco Systems](#) za svoj IOS (operativni sustav) koristi (Free)BSD Unix, dok za IOS XR koristi QNX Unix odnosno za NX-OS koristi [MontaVista Linux](#) itd. S druge strane [Juniper Networks](#) koristi FreeBSD Unix za sve svoje mrežne

uređaje. [Dell Networking](#) uređaji za pr. N seriju switcheva koriste Linux (DNOS 6.x OS), dok za S i Z seriju (koju su naslijedili kupnjom "Force 10" tvrtke) koriste FTOS koji je baziran na NetBSD Unixu ...

Nadalje i velika većina digitalnih telefonskih centrala (ako ne i sve) koristi neku varijantu UNIXa. Sve mobilne mreže ili barem vrlo veliki dio uređaja, servisa i opreme koji stoji iza njih također koristi neki UNIX. Pošto pristup internetu u konačnici završava kod nekog telekoma ili ISP-a, ponovno dolazimo do zaključka da bi "Internet" bez UNIXa postao upitan.

Ne bi bilo niti razvijene robotike: ne zaboravimo da su svi operativni sustavi u robotici ili direktno nasljednici nekog UNIXa ili su ideje njihovog razvoja potekle od njih. Najpoznatiji primjeri Unix-a koji pripadaju Tzv. kategoriji "[Real Time](#)" operativnih sustava (koji zadovoljavaju potrebe robotike koja je najčešća u industriji) su

- [QNX](#) (Danas "Black Berry") - pr. BlackBerry 10 je baziran na QNX Realtime UNIXu
- [VxWorks](#)
- i deseci drugih Realtime OS-ova.



Real Time operativni sustavi odnosno operativni sustavi za rad u stvarnom vremenu se znatno razlikuju od "običnih" operativnih sustava. U njihovom radu je vrlo važno vrijeme odziva određenog programa ili u konačnici procesa. Ovakvi operativni sustavi koriste se za vremenski kritične ili zahtjevne operacije, koje se moraju odraditi unutar točno definiranih vremenskih okvira. S druge strane kod "običnih" operativnih sustava, pojedini program odnosno proces dolazi na obradu (u CPU) u određenim vremenskim okvirima koji nisu toliko precizno definirani (obično ovise o prioritetu samog programa odnosno procesa) te se nikada ne može precizno znati u kojem trenutku će biti pokrenut proces obrade i kad će završiti.

Kod obje vrste operativnih sustava, komponenta sustava koja je zadužena za baratanje s programima odnosno procesima (pokretanje, pauziranje ili stopiranje) zove se "Task scheduler" ili "Process Scheduler". Za više detalja o njemu pogledajte knjigu "Uvod u Linux i Linux napredno" : poglavlje "[Napredno: Task scheduler](#)"

Za "Real Time OS" odnosno skraćeno RTOS ova vremena preuzimanja procesa i njegove obrade su ključna, stoga ne smije biti kašnjenja.

Uzmimo za primjer RTOS koji upravlja s robotskom rukom u proizvodnji automobila. Robotska ruka ima na desetke elektromotora i osjetnika (senzora), kojima upravlja RTOS. U ovakvom radu je vrlo važna spoznaja unutar kojih vremenskih okvira se koji motor mora pokrenuti i do koje pozicije. Također je važno u kojim intervalima se očitavaju vrijednosti s osjetnika. Ako bi ovi vremenski okviri bili nedeterministički, vrlo brzo bi došlo do neke havarije, jer bi se krivi motor aktivirao u pogrešnom trenutku ili bi program (proces) koji je zadužen za pokretanje motora morao čekati previše dugo na program (proces) koji je zadužen za osjetnike, te doveo robotsku ruku u krivi položaj. Za neke još zahtjevnije primjene posljedice loše koordiniranog rada bi bile još katastrofalnije.

Osim robotike, UNIX je u upotrebi i u avionskoj i automobilskoj industriji, kao i u medicinskoj industriji i mnogim svemirskim programima (od letjelica, orbitera do raznih satelita).

Lista upotrebe neke od inačica ili varijanti UNIXa je prilično velika, važno je samo shvatiti da bi svijet bez njega (UNIXa) bio potpuno drugačiji - vjerojatno bi prema upotrebi tehnologija koje bi nam bile dostupne bez UNIXa, danas živjeli u 70-tim godinama.

Vratimo se u povijest razvoja Unixa

Projekt GENIE

Projekt **GENIE** je bio aktivan od 1964 do 1967 i unutar njega je razvijen operativni sustav imena [Berkeley Timesharing System](#) koji je imao još neke napredne ideje i tehnologije koje su bile uzor za kasniji razvoj UNIXa.

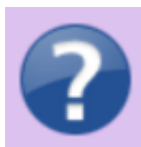
Naime iz projekta GENIE je razvijen *memory management* i pripadajuća logika koja je integrirana u postojeći hardver **SDS-930**, što je hardverski omogućilo mehanizme na razini operativnog sustava, koji su omogućavali sustav virtualne memorije. Razvijene su i slijedeće tehnologije i alati :

- Prvo je razvijena mogućnost pisanja programa u strojnom jeziku (Pr. Assembler)
- Potom je razvijen načini rada sustava u kojemu svaki korisnik može raditi u izoliranoj okolini ("separate protected user modes")
- Te vezano za sustav *virtualne memorije* direktno :
 - Cjelokupni "[Memory Paging](#)" sustav.
 - "[Protected memory](#)" sustav koji se brine o tome da niti jedan pokrenuti program (proces) ne može pristupiti memoriji koja nije alocirana za njega od strane sustava. Ovo je vrlo važan sigurnosni mehanizam bez kojega bi svaki program (proces) mogao doći do memorije koja mu nije namijenjena (dozvoljena) pa bi stoga mogao mijenjati i druge programe (proces) ili čak komponente operativnog sustava i izazvati greške u radu cijelog sustava - namjerno ili slučajno.



SDS-940 je često naziv komercijalne verzije ove implementacije koja je uključivala i operativni sustav.

Ken Thompson je također radio na **SDS-940** za vrijeme boravka na sveučilištu Berkeley



Za detalje o "Paging" sustavu. pogledajte knjigu : "[Uvod u Linux : Paging](#)".

Krajem 1960 istraživačka tvrtka **Bell Labs** je bila uključena u projekt sa sveučilištem **MIT** i tvrtkom **General Electric** na razvoju "time-sharing system" operativnog sustava, koji je nazvan Multiplexed Information and Computing Service (**Multics**). On je omogućavao većem broju korisnika paralelan (istovremeni) rad, odnosno pristup "[Mainframe](#)" sustavu.

Mainframe sustavi su snažna i kompleksna računala koja su se koristila unutar vrlo velikih organizacija ili tvrtki za kritične i zahtjevne zadaće (aplikacije).

Pojam "Mainframe" je nastao od naziva "main frames" - nazovimo ga glavnim okvirom (ormarom ili ormarima) unutar kojih su se nalazile komponente poput glavne memorije (RAM), CPU (central processing unit) i svih ostalih komponenti. Ova riječ se veže za vrlo snažne sustave ali i prva komercijalna računala za poslovnu primjenu. Mainframe sustavi su se intenzivnije počeli razvijati tijekom 1960, te se razvijaju i koriste i danas.



Mainframe računala se koriste i danas za vrlo specifične namjene, recimo u bankarskim ili nekim drugim osjetljivim sustavima. Takav je primjer [IBM zEnterprise System](#) koji se razvija i koristi i danas

Slika prikazuje Mainframe sustav tvrtke IBM : 704 iz 1964 godine

Autor : Lawrence Livermore National Laboratory



Canadaya razvio novi operativni sustav.

Nezadovoljan napretkom u razvoju ovog sustava [Bell Labs](#) se povukao. [Ken Thompson](#) koji je radio kao programer u njihovom razvojnom odjelu a i tada je radio na razvoju **Multicsa**, odlučio je napisati (razviti) svoj operativni sustav. S obzirom da je i dalje imao pristup razvojnom okruženju Multicsa prvo je odsimulirao novi datotečni sustav te "paging" sustav za memoriju. Ubrzo je zaključio da mu je potreban hardver koji je znatno jeftiniji i pristupačniji. U Bell labsu je naišao na malo korišteni hardver [DEC PDP-7](#) koji je poslužio svrsi.

Na navedenom DEC PDP-7 hardveru, je dvojac [Ken Thompson](#) i [Dennis Ritchie](#), uključujući Rudda

Ovaj operativni sustav je uključivao:

- hijerarhijski datotečni sustav
- koncept procesa
- sustav device datoteka (vidljivih u današnjim UNIX i Linux sustavima unutar direktorija /dev/).

Osim toga razvili su CLI (command-line interpreter) te nekoliko uslužnih programa. Unutar dodatnih mjesec dana Thompson je implementirao cijeli sustav u programskom jeziku [Assembler](#). Novi operativni sustav, koji će kasnije postati UNIX, sadržavao je assembler (programski jezik), osnovni tekst Editor i Shell (CLI).

Nevjerojatan dvojac: Ken Thompson i Dennis Ritchie, odlučio je zapravo, promijeniti svijet, ne prihvaćajući kompromise i postojeća rješenja.



Zapitajmo se što bi bilo s razvojem *Unixa*, za početak, da je navedeni dvojac, bez pogovora, slušao svoje "autoritete" (koji su praktično odustali od razvoja), bez razmišljanja i vlastitog intelekta. Naime oni su se kao dobri znanstvenici i inženjeri, vodili i premisom "Uvijek preispituj autoritete". Dakle razmišljaj svojom glavom, ne dopusti drugima, mada i bili autoriteti bilo koje vrste (znanstveni, menadžerski i sl.), da zamagle ili obezvrjeđuju vlastite ili tuđe spoznaje, vizije i ideje.

... i zemlja je nekada bila ravna ploha, za sve autoritete tog vremena, zar ne ?.

1970 godina

Početkom 1970 [Peter G. Neumann](#) je ovaj projekt nazvao **Unics** (UNiplexed Information and Computing Service) za razliku od **Multicsa** (Multiplexed Information and Computer Services). Tako je novi projekt dobio ime a uskoro su uslijedila sredstva za razvoj i tranzicija imena u **UNIX**.

Vrlo brzo, razvojnom **Unix** timu su se pridružili:

- [Douglas McIlroy](#), koji je radio na razvoju "[Unix Pipeline](#)" mogućnosti, te kasnije na programima:
 - [spell](#), [diff](#), [sort](#), [join](#), [graph](#), [speak](#) te [tr](#)
 - [Joe Ossanna](#), koji je radio na razvoju Unixa i programu [nroff](#) - prvo u **Assembleru** a kasnije u **C** jeziku.
- Ideja oko *Multics* projekta, odnosno njegovog dijela koji se bavi procesima (taskovima/zadacima) je bila u ranoj fazi razvoja ali se razmišljalo o tome kako je moguće izvoditi više zadaća istovremeno. S druge strane u rane verzije UNIXa, ugrađeni su drugačiji mehanizmi, kojima je cilj bio izvršavanje jednog po jednog zadatka, vrlo pouzdano i brzo.

Osim obrade procesa odnosno pojedinih zadataka na najnižoj razini (koje danas odrađuje task odnosno proces scheduler) i cijela osnovna organizacija datotečnog sustava, preuzeta je od strane "[Multics](#)" projekta.

Dodatno preuzete su i ideje koje su se razvijale od strane drugih ranih operativnih sustava, kao što je bio Massachusetts Institute of Technology's (MIT) CTSS. Zanimljivo je da je i sama ideja **fork** operacije kod kreiranja novog procesa došla od Berkeley [GENIE](#) projekta koji je razvijan na [SDS-940](#) hardveru.

Vratimo se na fork

Ideja **forka** je omogućila kreiranje i izvršavanje procesa bez zauzimanja puno resursa i to na način da se pokreće jedan proces za jednu naredbu za razliku od Multics-a u kojemu je bilo potrebno naredbe izvršavati kao podrutine (procedure), što je znatno kompliciralo rad i usporavalo sustav.



U UNIX-u (i Linuxu) svaki puta kada se kreira novi proces, **forka** (instancira i djelomično kopira) se trenutni proces tako da se kreira "dijete" proces od "roditeljskog" procesa. Dakle proces *forkanja* kopira trenutni proces odnosno sve njegove potrebne dijelove - bez podataka. Potom se pokreće novi proces (dijete) te mu se dodjeljuje novi PID (Process ID).

Ova procedura se pokazala vrlo brza jer kernel koristi "copy-on-write" sustav tako da je sve što je potrebno zapravo kreiranje *virtual memory mape* (tablice). Prema tome zapravo nema kopiranja memorije već se radi samo o pokazivačima na već postojeće memorijske adrese.

Nadalje veći dio ove procedure se odrađuje u MMU (Memory Management Unit) dijelu procesora - dakle događa se hardverskom brzinom.



Da ne bi bilo zabune ova procedura (**fork**) je štedljiva i što se tiče RAM memorije. Ne zaboravimo specifikaciju hardvera DEC PDP-11 poslužitelja :

" The PDP-11/45 on which our UNIX installation is implemented is a 16-bit word (8-bit byte) computer with 144K bytes of core memory; UNIX occupies 42K bytes. "

Izvor : "The UNIX TimeSharing System " : Dennis M. Ritchie and Ken Thompson , Bell Laboratories"

Programski jezik C

Paralelno s razvojem UNIXa , Dennis Ritchie i Ken Thompson krenuli su i s razvojem programskoj jezika **C**. Pošto je originalna inačica UNIXa razvijena na PDP-11, u Assembleru, ovaj dvojac je odlučio preći na programski jezik **B**, koji je ranije razvio Ken Thompson uz pomoć kolege Dennisa Ritchiea. Pošto programski jezik **B** nije mogao iskoristiti sve prednosti hardvera PDP-11 te je imao i neke druge nedostatke, odlučili su razviti i novi programski jezik, koji je nazvan **C**. Intenzivniji razvoj **C** jezika je krenuo 1972. godine, na UNIXu koji se pokretao na PDP-11 hardveru.

C jezik u početku nije razvijan s idejom portabilnosti na drugi hardver ali je ona vrlo brzo postala potrebna. Ubrzo nakon toga, **C** jezik, je portan (prenesen/razvijen) i na druge platforme : Honeywell 6000 a kasnije i na IBM System/370. Iste godine veći dio UNIXa je također prepisan u **C** jeziku te je već tijekom 1973 i gotovo cijeli UNIX kernel prepisan u **C** jeziku.

Negdje do 1977, Ritchie i Stephen C. Johnson napravili su dodatne modifikacije unutar **C** jezika koje su omogućile jednostavniju portabilnost i samog UNIXa na druge platforme.



Isti dvojac (Ken Thompson i Dennis Ritchie), vođen inženjerskim instinktom i načinom razmišljanja, ne pristaje na kompromise, već radi ono što bi i svaki izvrstan inženjer **morao** raditi : razvijati kvalitetne, robusne, pouzdane i sigurne proizvode.

Bilo bi dobro sve napraviti iz početka, ali prvo je potrebno razviti novi programski jezik (**C**) pa u njemu, od nule, napisati cijeli **Unix**. Ovakav problem je svakom izvršnom inženjeru, samo izazov i poticaj.

Jeste li inženjer ili ?



[Dennis Ritchie](#) zaslužan je za razvoj:

- [ALTRAN - nadogradnje na programski jezik FORTRAN](#)
- Programskog jezika [BCPL](#)
- Programskog jezika [B](#)

- [Multicsa](#)
- [Unixa](#)
- Programskog jezika [C](#)



[Kenneth Thompson](#) zaslužan je za razvoj:

- [Multicsa](#)
- [Unixa](#)
- Programskog jezika [BCPL](#)

- Programskog jezika [B](#)
- Programskog jezika [C](#)
- Programskog jezika [Go](#)
- [UTF-8](#) standarda

Obratite pažnju na Ritchieja i Thompsona - razvoj cijelog operativnog sustava bez monitora kao izlaznog uređaja je u to vrijeme bio normalan (postojao je samo ispis na štampač).



- Probajte prvo razviti cijeli operativni sustav u programskom jeziku **Assembler** (što obuhvaća tisuće redova koda) - bez monitora
- Potom napisati programski jezik **C**, također u **Assembleru**, s još pokojom tisućom redova (linija) koda, također bez monitora
- Zatim cijeli operativni sustav iz početka prebaciti (prevesti) u programski jezik **C**

Sve navedeno je samo djelić problema s kojima su se susreli

Mislim da sada polako shvaćate razinu entuzijazma i altruizma ovih ljudi. Oni bi trebali biti školski primjer pravih znanstvenika i inženjera, koji mijenjaju svijet na bolje. Uvijek propitkujući i tražeći neko bolje i kvalitetnije rješenje. Nikada ne prihvaćajući prosječno ili osrednje. Ovo su stvarno veliki ljudi, na čije djelo iznova i iznova treba podsjećati. Oni bi trebali biti uzor svim generacijama znanstvenika i inženjera.

Autor slike :Peter Hamer - Ken Thompson (sitting) and Dennis Ritchie at PDP-11. Uploaded by Magnus Manske, CC BY-SA 2.0, <https://commons.wikimedia.org/w/index.php?curid=24512134>

Godine 1978, [Brian Kernighan](#) i Dennis Ritchie napravili su prvo izdanje knjige o **C** programskom jeziku znano i kao "K&R" C. Drugo izdanje ove knjige poriva kasnije verzije C jezika znanog i kao **ANSI C**.



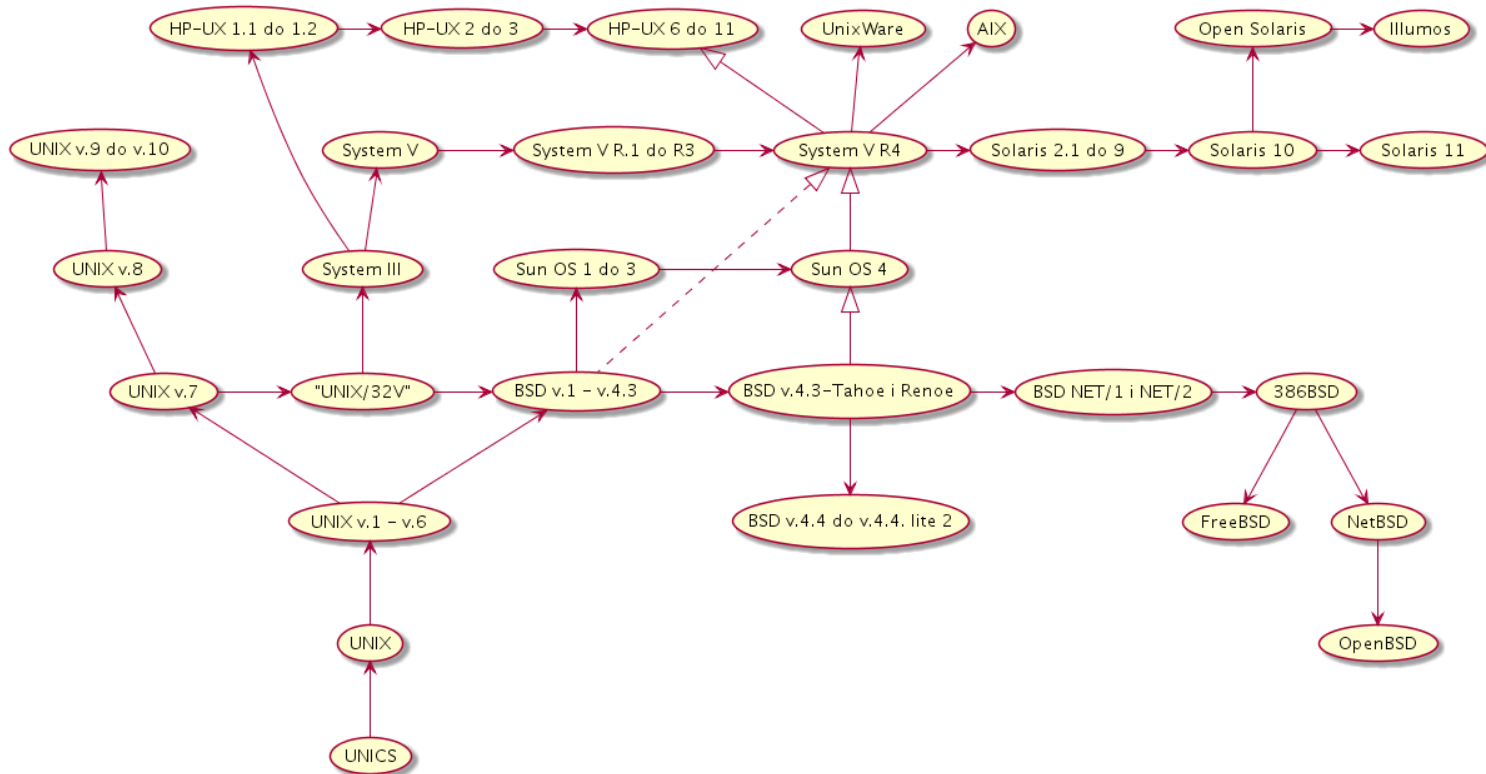
[Brian Kernighan](#) zaslužan je za razvoj:

- Unixa
- Programskog jezika **AWK**
- Programskog jezika **AMPL**

Podrijetlo vrsta

U tekstu koji slijedi, pratiti ćemo razvoj UNIXa koji je doveo i do razvoja FreeBSDa. Dakle samo one grane u UNIX stablu koje će nam biti zanimljive u svrhu ovog članka. Također ćemo preskočiti neke razvojne faze i verzije (a i godine) u kojima se nije događalo ništa nama zanimljivo.

Pogledajte koliko je razvoj raznih inačica, o kojima ćemo govoriti, bio isprepleten (ovo je samo mali segment cijele priče o Unixu):



Prve od važnijih inačica UNIXa su nastale kao razvojne unutar tvrtke **Bell Laboratories** te su sadržavale **UNIX Time-Sharing System** implementaciju, i to šestu inačicu, znanu i kao **V6**. Inačica **V6** je tijekom 1976 bila i prva široko prihvaćena izvan tvrtke **Bell Laboratories**.



Inačica **V6** je bila široko prihvaćena iz razloga što je bila razvijana u višem programskom jeziku (**C**), koji je pružao mogućnost brzog razvoja, podršku za nove funkcionalnosti hardvera i dodatno što je bio objavljen njen programski kod. Osim toga imala je implementirana neka riješenja koja su do tada bila rezervirana samo za znatno kompleksnije sustave.

Ovaj potez davanja kompletnog izvornog koda UNIXa rezultirao je njegovim nevjerojatnim širenjem, dodavanjem novih funkcionalnosti, optimizacijama i sl.

Još jedna važna stvar (u to vrijeme) je bila mogućnost UNIXa, za korisnike, pokretanje više procesa (programa) istovremeno, kao i mogućnosti povezivanja procesa (naredbi) kroz " [pipe](#) " mogućnost. "Pipe" - funkcionalnost je uvela mogućnost pokretanja naredbi tako da se rezultat jedne naredbe šalje kao ulaz drugoj naredbi i tako dalje, stvarajući nove funkcionalnosti, bez potrebe za razvojem potpuno novog programa.

Slika: DEC PDP-11



Autor slike: Stefan_Kögl <https://commons.wikimedia.org/w/index.php?curid=466937>

U to vrijeme takve stvari su bile nezamislive jer su samo najveći sustavi mogli pokretati više istovremenih procesa a broj procesa koji je pojedini korisnik uopće mogao pokrenuti je bio ograničen od strane sistem administratora.

Većina ranih inačica UNIXa izvršavala se na [PDP-11](#) hardveru, koji je bio cjenovno prihvatljiv a i dovoljno snažan.

Mana ovog hardvera je bila vrlo mali adresni prostor.

Slika: VAX11/780



Razvojem strojeva s 32-bitnim adresnim prostorom a posebice [VAX-11/780](#) omogućila je novim inačicama UNIXa proširenje mogućnosti te se uveo sustav virtualne memorije (pogledajte "[Uvod u Linux](#)" i [virtualnu memoriju](#)).

S obzirom da se pojavljivao novi i snažniji hardver a i činjenicu što se davno s **Asemblera** prešlo na **C** jezik, portabilnost na različite arhitekture više nije bila toliki problem.

Prva inačica UNIXa kojoj je portabilnost na različite hardverske platforme bila prioritet bila je "UNIX Time-Sharing System, Seventh Edition (V7)". Ona se sada mogla izvršavati na PDP-11 ali i na Interdata 8/32 te je imala inačicu koja se mogla izvršavati i na VAX sustavima - ova inačica se zvala: "UNIX/32V TimeSharing, System Version 1.0 (32V)".

Autor slike : Emiliano Russo, Associazione Culturale

VerdeBinario http://it.wikipedia.org/wiki/Immagine:VAX_11-780_all.jpg



Istovremeno je AT&T licencirao svoju inačicu **5** edukacijskim institucijama a inačicu **6** i komercijalnim korisnicima. Škole su plaćale licencu 200 U\$ a svi ostali 20.000 U\$.

Tvrtka **Bell Laboratories** je još razvila inačice:

- **V8** (1985.g.)
- **V9** (1986.g.) i
- **V10** (1989.g.)



Tvrtku **Bell Labs** je osnovao Alexander Graham Bell, krajem 19-stog stoljeća.

Kasnije nastala tvrtka **Southwestern Bell Corporation** je osnovala tvrtku **AT&T**.

Tvrtka **Bell Laboratories** je danas (2016 -) u vlasništvu tvrtke "Nokia".

AT&T UNIX System III i UNIX System V

Nakon distribucije Sedme verzije (V7) UNIXa 1978, razvoj je prebačen na grupu "UNIX Support Group (USG)" koja ga nastavlja razvijati.

AT&T je najavio **System III** krajem 1981 godine kao prvo izdanje koje nije razvijano unutar *Bell Labsa* već unutar USG-a. Ova USGova prva inačica (nakon sedme inačice UNIXa (V7)) nazvana je **UNIX System III** (3) koja je razvijena tijekom 1982- 1983 godine. Ona je zapravo bila kombinacija raznih AT&T UNIXa : PWB/UNIX 2.0, CB UNIX 3.0, UNIX/TS 3.0.1 i UNIX/32V.

Unix System III je podržavao DEC PDP-11 i VAX računala (hardverske platforme).

Ova inačica UNIXa je nazvana ovako (III) jer se smatrala vanjski izdanjem koje se većinom naslanjalo na UNIX/TS 3.0.1 CB UNIX 3, a koje je razvijao *Bell Labs*.

Unix System I i Unix System II nikada nisu postojali

Unix System III (3) je između ostaloga uveo "named pipes" , "uname" sistemske pozive i naredbe i "run queue" uz još mnoga druga poboljšanja i nove funkcionalnosti.



Za više informacija - pogledajte knjigu **Uvod u Linux i Linux napredno** - poglavlja :

- ["Named pipes"](#)
- ["Osnovne naredbe vezane uz operativni sustav i komponente računala"](#)
- [Task Scheduler](#)



Pojam **Research Unix** se odnosi na sve inačice *Bell Labs* UNIXa koji se pokretao na platformama :

- DEC PDP-7
- DEC PDP-11
 - VAX
- Interdata 7/32 i Interdata 8/32

Odnosno koji se razvijao unutar *Bell Labs* Computing Science Research Center (CSRC), za navedene platforme.



Naziv "**Research Unix**", prvi puta se pojavio tek u Unix inačici 8 (**V8**) te se često u literaturi koristi i za sve ranije verzije.

Unix System V (5)

USG je ubrzo (1983 godine) izdao Unix System V (čita se pet) koji je bio nasljednik Unix System III.

Potom je USG razvojna grupa službeno postala UNIX System Development Laboratory (USDL). Zatim su izdali podverziju 2 u 1984 godini (Unix System V Release 2) poznatu kao **SVR2**.

USDL je zamijenjen s AT&T Information Systems ([ATTIS](#)), koji je sada distribuirao UNIX System V, Release 3 (SVR3), 1987 godine.

Kasnije je ATTIS postao [Unix System Laboratories](#) (USL) - od 1988. godine

System V Release 3 (SVR3) je uvela

- **STREAMS** - koji je adaptiran iz V8 inačice UNIXa - on je *framework* za implementaciju *character device driver*a, mrežnih protokola i IPC sustava ([Inter Process Communication](#)). **IPC** sustav je vrlo važan u svakom operacijskom sustavu, jer je zadužen za komunikaciju između programa (procesa). Dakle sva komunikacija programa među sobom prolazi kroz ovaj sustav.
- **Remote File System (Sharing)** (RFS) - koji je bio preteča **NFSa** (Network File Share)
- **File System Switch** (FSS) , koji je praktično isto bio preteča **Virtual File System** (VSS) sustava, koji se koristi i danas. Naime **Virtual File System** je veza između korisnika odnosno korisničkih programa i datotečnog sustava, bez obzira kakvi oni bili - po vrsti odnosno tipu ili lokaciji (lokalno ili preko mreže). S druge strane, on je spoj prema kernelu i upravljačkim programima. Kao primjerice za disk kontroler i tvrde diskove u konačnici - ili za mrežne diskove (pr. mrežni **NAS** sustav).
- i mnoge druge stvari

Ovdje su razvijene razne tehnologije koje su kasnije iskorištene i za druge (novije) inačice UNIXa

System V Release 4 (SVR4):

Tijekom 1988 te 1989. godine **USL** i **Sun Microsystems** su razvili novu inačicu (4) **Unix System V**, uz upotrebu najboljih tehnologija iz drugih Unixa.

System V Release 4 (SVR4) je uveo:

- **paging** odnosno upotrebu "stranica" unutar sustava virtualne memorije (pogledajte dolje)
- **Copy-on-Write** mehanizam unutar sustava virtualne memorije - ekstremno važan kod kreiranja procesa (sjetite se **fork** mehanizma kod pokretanja programa/procesa) te iznimno korišten u radu sustava
- sustav dijeljenja memorije (**shared memory**) unutar sustava virtualne memorije.



Ova tri mehanizma unutar sustava virtualne memorije su vrlo važna i danas se koriste na svim operativnim sustavima.

Paging o kojemu stalno govorimo vrlo je važan element sustava virtualne memorije. Pojednostavljeno on radi na principu u kojemu se virtualna memorija dijeli u blokove iste veličine, koji se nazivaju *pages* ili *stranice* virtualne memorije. Svakim pristupom (virtualnoj) memoriji se zapravo pristupa u blokovima odnosno stranicama, kako za čitanje iz memorije tako i za zapisivanje u nju. Slično kao što se na diskovima podaci zapisuju u blokove koji se obično zovu **sektori**.

Za detalje pogledajte knjigu "Uvod u Linux i Linux napredno" : "[Virtualna memorija](#)"

Osim navedenog u ovu inačicu (System V R4) su uključene i razne napredne tehnologije.

Iz **BSD Unixa**:

- **TCP/IP** protokol
- Unix [Socketi](#) - vrlo važnu tehnologiju za rad bilo kojeg Unixoidnog sustava - pogledajte *Berkeley sockets* pred kraj knjige.
- **UFS** ([Unix File System](#)) datotečni sustav
- [C Shell](#) i drugo

Iz **SunOS Unixa**:

- **Virtual File System** ([VFS](#)) , koji je zamijenio stari *File System Switch* o kojem smo govorili - a koji se koristi i danas.
- **NFS** ([Network File System](#)) mrežni dijeljeni datotečni sustav, koji se i danas koristi.
- Napredni sustav dijeljenih biblioteka ([Shared Library](#)) - koji se također koristi u svim operacijskim sustavima.
- **OpenWindows GUI** - grafičko sučelje
- i drugo

Iz **Xenix Unixa**:

- Upravljački programi za x86 arhitekturu procesora te binarna kompatibilnost sa *Xenixom*.

Dodana je i podrška za:

- [Korn Shell](#)
- **Application Binary Interface (ABI)** baziran na Tzv. *Executable and Linkable Formatu (ELF)* - također vrlo važna komponenta za sve današnje *Unixoidne* operativne sustave (uključujući Linux)
- Standarde, poput [POSIX](#) i [X/Open](#).
- [ANSI \(X3J11\) C](#) programski jezik. Naime do tada su se koristile razne implementacije programskog jezika **C**. X3J11 je postavio standarde za programski jezik *C* za implementaciju *Unixa*.

Za **SVR4** inačicu *Unixa* možemo reći da je definitivno postavila nove standarde, uz uključene sve najvažnije tehnologije. Ova verzija je *De facto* standard, sama po sebi.



Od **SVR4** inačice *Unixa* su potakle mnoge varijante *Unixa*, koje su počele razvijati razne tvrtke :

- Atari - [Atari System V](#)
- Commodore - [Amiga Unix](#)
- Fujitsu - [UXP/DS](#)
- Hewlett-Packard - [HP-UX](#)
- IBM - [AIX](#)
- NCR - [Unix/NS](#)
- NEC - [EWS-UX](#)
- SGI (Silicon Graphics) - [IRIX](#)
- Siemens - [SINIX](#)
- Sony - [NEWS-OS](#)
- Sun Microsystems - [Solaris](#)
- i mnoge druge

init skripte (SysV init)

Unix System V je unio još jednu važnu stvar u Unix a kasnije i Linux svijet. Radi se o inicijalizacijskim shell skriptama (init skriptama) koje se pokažu kod pokretanja sustava (i gašenja naravno).

Dakle AT&T UNIX System III je uveo novi način pokretanja sustava odnosno pokretanja inicijalizacijskih shell skripti, a koji je naslijeđen i u Unix System V (uz određene modifikacije).

Zbog činjenice što je način inicijalizacije sustava pomoću init skripti iz Unix System V postao standard, koji su slijedile i mnoge kasnije verzije Unixa kao i razne distribucije i Linuxa, on je nazvan **"SysV-style init"**.

Naime u bilo kojem trenutku Unix Sytem V (i svi oni koji koriste ovakav **SysV-style init** način rada) nalazi se u jednom od stanja rada, poznatih pod imenom **runleveli**. Svaki sustav ima definirano minimalno nekoliko runlevela ili stanja rada, od kojih je barem jedan "normalan" odnosno onaj koji omogućava višekorisnički rad s punim radnim okruženjem, mrežom i svim potrebnim servisima (daemonima).



Normalna stanja rada sustava su obično :

- **runlevel 3** - višekorisnički rad u tekstualnom načinu rada s mrežom i svim pokrenutim servisima i mrežom
- **runlevel 5** - višekorisnički rad u grafičkom (X Window) načinu rada s mrežom i svim pokrenutim servisima i mrežom, na nekim System V sustavima ovo je slično runlevel 0 uz odlazak u "Firmware" poslužitelja.

Osim navedenih "normalnih" postoje i drugi načini rada koji su obično rezervirani za jednokorisnički rad i to bez raznih servisa, kao i bez mrežnih diskova i sl. - pa se obično koriste za popravljjanje ili održavanje sustava (**runlevel 1**). I na kraju postoje i runleveli kojima je jedina namjena uredno gašenje svih servisa (daemoni) te pripremanje sustava za konačno gašenje te runlevel za restart sustava.



Ostali runleveli koji su često u upotrebi:

- **0** Halt (gašenje sustava)
- **6** Reboot (restart sustava)

U konačnici se rad korištenjem System V kompatibilnih **init** skripti svodi na pokretanje niza skripti koje su definirane za pokretanje na razini svakog načina rada (runlevela). Dakle svaki pojedini runlevel ima točno definirane skripte koje se pokreću prema definiranom redoslijedu.

Unutar vršnog direktorija `/etc/rc.d/` se nalaze poddirektoriji koji već i imenom predstavljaju **runlevele** :

- `rc0.d/`
- `rc1.d/`
- `rc2.d/`
- `rc3.d/`
- `rc4.d/`
- `rc5.d/`
- `rc6.d/`

Unutar njih se nalaze gore navedene datoteke - specifične za svaki od runlevela.



Za više informacija o **initu** i **init skriptama** pogledajte knjigu **"Uvod u linu i Linux napredno"**, poglavlje ["Runlevels"](#)

Berkeley Software Distributions (BSD)

Ponovno ćemo se vratiti u povijest jer ova priča počinje negdje u vrijeme razvoja prvih **AT&T** odnosno **Bell Labs**ovih inačica Unixa.

Profesor [Bob Fabry](#) koji je radio na kalifornijskom sveučilištu [Berkeley](#), pokrenuo je nabavku (kupnju) UNIX Source licence od AT&T-a. Ova licenca je obuhvaćala izvorni kod njihovog UNIXa. Plan je bio modifikacija postojeće verzije AT&T UNIXa, prema njihovim željama odnosno planovima. Tako je nastao **BSD Unix** (Berkeley Software Distribution).

Zanimljivo je i to da je 1975 godine, Ken Thompson “uzeo slobodno” od Bell Labsa i došao na sveučilište Berkeley kao gostujući profesor. On je pomogao pri instalaciji AT&T UNIXa verzije 6 te je krenuo s implementacijom programskog jezika [Pascal](#) na istu platformu. Studenti Chuck Haley i [Bill Joy](#) poboljšali su Thompsonovu verziju Pascala te implementirali (tada) napredni tekst editor [ex](#).

1BSD (za PDP-11)

S obzirom da su i druga sveučilišta postala zainteresirana za rad na ovoj inačici Unixa, Joy je krenuo s kompiliranjem prve verzije “**Berkeley Software Distribution**” (1BSD), koja je objavljena početkom 1978.

Ova rana inačica BSD Unixa je više bila nadogradnja AT&T Verzije 6 (V6) Unixa nego zaseban operativni sustav.



Bill Joy je jedan od ranih osnivača tvrtke [Sun Microsystems](#). Osim toga autor je popularnog [vi](#) tekst editora (1976.g.) koji je inicijalno bio nadogradnja na njegov (njihov) **ex** editor.

Bio je uključen i u razvoj:

- [NFS](#)-a (Network File System)
- Programskog jezika [JAVA](#)
- C Ljuske ([C Shell](#))
- TCP/IP protokola
- [chroot](#) funkcionalnosti
- [SUN Solaris](#) operativnog sustava

Vratimo se na početke razvoja BSDa

2BSD (za PDP-11)

Druga inačica BSD UNIXa (znana kao **2BSD**) dovršena je i objavljena 1979. godine. Sadržavala je nadograđenu verziju 1BSDa te dva dodatna programa koja je razvio Bill Joy :

- **vi** tekst editor (“grafička” nadogradnja na **ex** tekst editor)
- **c shell** odnosno “c ljusku” (CLI).

Druga važna i nova funkcionalnosti koje je 2BSD imao je mrežni paket (softvera) Tzv. “Berknet”, koji je razvio [Eric Schmidt](#) u sklopu svog diplomskog rada. Ovaj mrežni paket softvera je donio mogućnost umrežavanja čak do 26 računala te je uključivao e-mail te mogućnost razmijene datoteka preko mreže - vrlo napredno za to vrijeme.

Uslijedile su i druge podinačice verzije 2, od kojih su novije koristile i neke funkcionalnosti novijih grana BSD-a:

- 1982 godine inačica **2.8.1** BSD
- 1983 godine inačica **2.9** BSD (ova inačica primjerice koristi kod iz inačice 4.1c BSD)
- 1992 godine inačica **2.11** BSD

Paralelni razvoj verzije 3 BSD-a

Ovom inačicom se odlučilo preći na novi hardver i uvesti neke druge promijene. Ovdje je kao baza bio odabran **UNIX/32V** kojemu je nedostajala mogućnost upotrebe sustava virtualne memorije koji je bio dostupan na VAX sustavima. Stoga je veliki dio ovog kernela morao biti ponovno napisan (prepisan) da bi se dobila ova važna funkcionalnost. Uslijedio je razvoj ranih programa i alata s verzije 2BSD za ovu novu VAX platformu te je nastao 3BSD, negdje tijekom 1979 godine.

4BSD

Na osnovi inačice 3BSD, koja je bila široko prihvaćena a koja je i primijećena od strane DARPA agencije, razvijena je i inačica 4BSD, krajem 1980.

Istovremeno, negdje sredinom 1980 profesor Fabry je potpisao ugovor s [DARPA](#) (Defense Advanced Research Projects Agency) agencijom za potrebe [ARPAnet-a](#).



Advanced Research Projects Agency Network (ARPANET) je bio prvi projekt kojemu je jedan od ciljeva bio umrežavanje računala lokalno i na velikim udaljenostima pomoću Tzv. *packet switchinga*. On je kasnije postao i prva mreža u kojoj je implementiran [TCP/IP](#) niz protokola. Obije ove ideje : *packet switching* kao i razvoj TCP/IP-a su postale osnova današnjeg interneta.

S osiguranim novčanim sredstvima DARPA agencije osnovan je "Computer Systems Research Group" ([CSRG](#)) unutar kojeg će se nastaviti razvoj te koji će distribuirati budući BSD UNIX.

Sada je osnovni cilj razvoja cijele generacije 4 BSD Unixa postao podrška za DARPA-in "Internet networking protocol" odnosno TCP/IP set protokola. Vrlo brzo implementacija ovog mrežnog protokola omogućila je komunikaciju uređaja (poslužitelja) koji su bili u lokalnim mrežama, povezanim ili "Token Ring" ili "Ethernet" mrežama te komunikaciju preko "WAN" mreža (koje pokrivaju velike udaljenosti) kao što je bio DARPA-in "ARPANET".

Uslijedile su razne inačice:

- **4.0BSD**
- **4.1BSD** (1981.g.)
 - **4.1a** preliminarne [BBN](#) implementacija TCP/IP-a
 - **4.1b** implementiran "[Berkeley Fast File System](#)", znan i kao Unix File System (UFS), od strane [Marshall Kirk McKusicka](#), jednog od najpoznatijih ikona BSD-a.
 - **4.1c** - ova inačica je postala osnova za razvoj *Bell Labsovog* "Research Unix"-a verzije osam (V.8.).
- **4.2BSD** (1983.g.) Na ovu inačicu je značajno utjecalo DARPA povjerenstvo. Dodatno su implementirana mnoga poboljšanja, među kojima je uvođenje "disk quota" i "job control" mehanizama. [Mike Karels](#) i [Marshall Kirk McKusick](#) preuzimaju vodstvo u upravljanju daljim razvojem.
- **4.3BSD** (1986.g.) : Nakon nekoliko mjeseci intenzivnog testiranja TCP/IP dijela, od strane DARPA agencije, utvrđeno je da je TCP/IP iz 4.2BSD verzije superioran u odnosu na druge te da on ostaje unutar nove verzije 4.3.BSD.



[Marshall Kirk McKusick](#) je jedan od najaktivnijih članova BSD zajednice. Uključen je u rani razvoj BSDa od 1980tih sve do danas (do FreeBSDa). Zaslužan je za razvoj:

- [Berkeley Fast File System](#) datotečnog sustava, iz kojega je kasnije nastao [UFS](#) datotečni sustav, jedan od najkorištenijih na raznim Unix operacijskim sustavima
- [Soft updates](#) - metode za održavanje i provjeru datotečnog sustava
- **UFS2** Datotečnog sustava
- [Filesystem Snapshot](#) - metode snimanja stanja u vremenu na datotečnom sustavu
- [fsck](#) alata za pronalazak i popravljivanje grešaka na datotečnom sustavu
- i mnoge druge komponente



Možda biste pomislili, pošto se radi o državnoj agenciji (**DARPA**), kako je njihova stručnost ili posvećenost zadaćama koje su dobili, za procjenu implementacije TCP/IP protokola upitna, ali varate se. Naime u DARPinom odboru su sve redom bili vrhunski stručnjaci:

- [Bob Fabry](#)
 - [Bill Joy](#) - za više informacija o njegovim dostignućima pogledajte par stranica gore.
 - [Samuel J Leffler](#) (Autor: FlexFAX, libtiff i upravljačkih programa za Wireless za FreeBSD) sa [UCB](#) (University of California, Berkeley).
 - Alan Nemeth
 - Rob Gurwitz iz tvrtke [BBN](#), gdje je radio kao programer na TCP/IP protokolu.
 - [Dennis Ritchie](#) tada iz tvrtke *Bell Labs*
 - Keith Lantz sa sveučilišta [Stanford](#).
- Bili su tu još i [Richard Rashid](#) (Autor mnogih patenata iz područja kompresije podataka, računalnih mreža i dr.) sa sveučilišta [Carnegie Mellon](#) te Bert Halstead s instituta [MIT](#). Nadalje Dan Lynch s instituta [ISI](#) te [Gerald J. Popek](#) sa sveučilišta [UCLA](#).

- Te podvarijante **4.3BSD Tahoe** (1988.g.) i
 - **Net 1** (1989.g.)
- **4.3BSD Reno** (1990.g.) i
 - **Net 2** (1991.g.) - nasljednik **Net 1**

Kulminacijom razvoja verzije 4.2 BSD-a na koju je najznatnije utjecala suradnja s DARPA agencijom (1983) nastala je inačica 4.3 BSD (sredina 1986. godine). Uslijedila je inačica 4.3 BSD Tahoe (1988.g.) te 4.3 BSD Reno (1990.g.).

U međuvremenu (između **Tahoe** i **Reno**) inačica pojavile su se verzije koje su jednostavno nazvane : 4.3BSD Net1 (početak 1989.g.) i 4.3BSD Net2 (sredina 1991.g.). Ove dvije verzije su imale zadatak izbacivanja kompletnog (svog) koda koji je bio "proprietary" odnosno pokriven "UNIX source" licencom.

Kako je VAX platforma zastarjevala, nova inačica 4.4BSD nije više razvijana za taj hardver.

Inačica 4.4BSD je stoga paralelno razvijena za nove platforme:

- **Motorola 68000**
- **SPARC**
- **MIPS**
- **Intel PC**

init skripte (BSD init)

Za razliku od "SysV-style init" skripti BSD koristi jednostavniji model u kojemu nema klasičnih **runlevela**.

Kod pokretanja sustava, učitava se prva skripta (`/etc/rc`) koja je zadužena za pokretanje svih ostalih inicijalizacijskih skripti. Ona čita dvije konfiguracijske datoteke (`/etc/rc.conf` i `/etc/defaults/rc.conf`) u kojima su definirane init skripte koje se moraju pokrenuti (u konačnici servisi/daemoni).

Nakon što je utvrđeno koje inicijalizacijske skripte se moraju pokrenuti, one se i pokreću. Sve init skripte se nalaze unutar direktorija `/etc/rc.d/` i `/usr/local/etc/rc.d/` slično kao što se init skripte kod System-V inita zapravo nalaze unutar `/etc/init.d/` te su kreirani simbolički linkovi na njih, a koji se nalaze negdje unutar nekog od direktorija `/etc/rc.d/rc.XY.d`, ovisno u kojem runlevelu se moraju izvršiti. Kod BSD inita to očito nije slučaj. Dakle nema strukture poddirektorija `/etc/rc.d/rc.XY.d` (XY je runelevel : 0-6) niti runlevela.

Kod BSDa se unutar direktorija `/etc/rc.d/` nalaze sve skripte koje pokreću sistemske servise (daemone) važne za rad operativnog sustava, a koje pripadaju u kategoriju Tzv "base system".

S druge strane postoji još jedan direktorij sa init skriptama za korisnički instalirane servise odnosno one koji ne pripadaju kategoriji "base system" a one se nalaze unutar direktorija `/usr/local/etc/rc.d/`. Ovdje se nalaze servisi poput:

- Apache Web poslužitelja ili nginx Web poslužitelja
- SNMP poslužitelja
- SQUID Proxy poslužitelja
- SSH poslužitelja (Open SSH)
- OpenVPN poslužitelja i sl.

Definicija servisa (init skripti) koji se moraju pokrenuti se nalazi u datoteci : `/etc/rc.conf` i izgleda otprilike ovako (pr. za SSH servis):

```
openssh_enable="YES"
```

Tranzicija BSD UNIXa u Open Source

Sve do verzije 4.3 Tahoe, svi korisnici BSD Unixa su prvo morali platiti **AT&T**-u "source" licencu. To je bilo stoga što BSD sustav nikada nije objavljen od strane Berkeley-a, samo u binarnom formatu, već je cijeli BSD Unix distribuiran sadržavajući cjelokupni izvorni kod kompletnog sustava odnosno svih njegovih dijelova.



Upravo zbog toga je i bio toliko prihvaćen (BSD), jer su mnoge institucije, kao i ogroman broj pojedinaca, aktivno pridonijeli njegovom razvoju, poboljšanjima, novim funkcionalnostima kao i otkrivanju i ispravljanu grešaka.

S vremenom su troškovi **AT&T**-ove "source" licence sve više rasli (klasična priča o pohlepi i managementu) te su svi oni željni razvoja svojih varijantu UNIXa baziranih na BSDu, a koji su poglavito koristili odlično razvijen TCP/IP i pripadajuće mrežne servise odustali od daljeg rada. Uz odustajanje sve više njih je tražilo od Berkeley-a odvajanje svog mrežnog koda (i za TCP/IP i sve pripadajuće servise) i izdavanja pod nekom drugom (normalnijom / manje pohlepnom) licencom.

Pošto TCP/IP mrežni kod nije niti postojao u inačici **32/V** (koji je bio pod AT&T licencom) već je u potpunosti bio razvijen od strane Berkeley-a i drugih pojedinaca i grupa, sve je i izašlo sredinom 1989 godine u novoj verziji **BSD Networking Release 1**, konačno prvaj slobodnoj od nameta i ograničenja.



Nešto kasnije (1993.g.) i **Microsoft** je u svoj **Windows NT v.3.1** uključio dobar dio ovog TCP/IP koda, a navodno se isti nastavio razvijati i za novije verzije *Windowsa*.

BSD licenca je bila liberalna. Svatko je mogao koristiti ili distribuirati izvorni ili promijenjeni programski kod ili binarno, bez plaćanja licence.

Jedini zahtjev je bio da se izvorni dokument s pravima upotrebe, autorima i sl. moraju distribuirati nepromijenjeno. Praktično se navodi da program sadrži izvorni kod Kalifornijskog sveučilišta Berkeley i pojedinaca koji su sudjelovali u razvoju.



BSD licenca je tzv. **non-copyleft** licenca, za razliku od recimo **GNU GPL** licence. To znači da ako program razvijate prema BSD licenci, programski kod **MOŽETE** objaviti u cijelosti ali imate i mogućnost da određeni dijelovi vašeg programa budu dostupni samo i isključivo u binarnom obliku. Binarni oblik podrazumijeva nedostupnost samog programskog koda. To implicira mogućnost da možete imati recimo program koji se sastoji od dvije datoteke, prvoj možete objaviti programski kod u cijelosti, a drugu možete kompilirati te ju distribuirati u binarnom obliku (iz kojega se programski kod ne može iščitati).

Na početku je Berkeley naplaćivao 1.000 \$ za traku na kojoj je bio pohranjen izvorni kod ali je svatko mogao kopirati kod od nekog drugoga tko je već kupio izvornu traku (medij s podacima). Čak se vrlo brzo pojavilo nekoliko internet stranica koje su preko FTP protokola omogućavale pristup samom kodu.

S druge strane nekoliko stotina organizacija je kupilo trake s izvornim kodom, što je znatno pomoglo u daljem razvoju i financiranju CSRG grupe.

Networking Release 2

Zbog uspjeha prvog **Networking Releasea**, CSRG, [Keith Bostic](#) (vodeći programer) je odlučio sve veći dio programskog koda prepisivati "od početka" tj. napraviti "rewrite" ostatka BSD Unixa, pod BSD licencom. Dakle razviti ga od početka i objaviti mu izvorni programski kod.

Prvo se krenulo s raznim uslužnim programima (UNIX utilities), koji su morali biti napisani od nule, ne koristeći niti liniju koda od strane **AT&T**a odnosno koristeći samo dostupne opise funkcionalnosti postojećih UNIX programa. Jedino što bi autori dobili bi bilo spominjanje njihovog imena u Berkeley licenci, za programski kod (program) koji bi napisali.



Iz početka sve je išlo vrlo sporo, te su napisani samo jednostavniji programi. S vremenom je lista počela rasti te je Bostic promovirao rad na javnim događajima "[USENIX](#)" grupe, koja je bila široko prihvaćena, što se pokazalo punim pogotkom. Sve više programera se uključilo i krenulo s razvojem.

Ubrzo se lista "odrađenih" programa povećala na 100 i nastavila je rasti te su unutar 18 mjeseci dovršeni svi važni programi i biblioteke.

Ostao je sam kernel koji je bio i najveći zadatak. Kernel je polako revidiran, datoteka po datoteka, te se polako izbacivao licencirani (AT&T Unix 32/V) programski kod te razvijao novi (slobodni) kod.

Nakon mjeseci rada, ostalo je još samo nekoliko datoteka koje je trebalo promijeniti. Međutim ipak je odlučeno da će se konačna inačica objaviti što prije, te je objavljena pod imenom **Networking Release 2**, sredinom 1991. godine (s time da je ostalo još šest datoteka koje nisu "odrađene").

Ponovno je nekoliko stotina pojedinaca i organizacija platilo po 1.000 U\$ kao potporu za razvoj.

386/BSD , NetBSD i FreeBSD

Berkeley je razvijao i "proprietary" odnosno "licenciranu" verziju od strane "Berkeley Software Design"-a (BSDi). Ova inačica je imala naziv **BSD/386**

William Jolitz, koji je do tada radio u BSDi, je sam odlučio odraditi i ostatak posla tada (onih šest datoteka) te je nakon 6 mjeseci rada izdao kod te sve kopirao i izradio kompletnu verziju operativnog sustava na 386 PC arhitekturi, koju je nazvao 386/BSD.

Nažalost zbog manjka vremena, William Jolitz se više nije mogao posvetiti daljem radu, zbog sve većih potreba i želja za razvojem, novim funkcionalnostima i sl.

BSDi je kasnije s **BSD/386** krenuo s prodajom licenci i podrške.

Prednost ove verzije (BSD/386) razvijane od strane BSDi je bila u vrlo opširnim i detaljnim uputama (man stranicama), od naredbi, do cijelog sustava. Odličnih objašnjenja sintakse, primjera i sl.

Ovaj sustav je kasnije nazvan **BSD/OS**.



U ovoj priči je došlo i do tužbe znane kao [USL vs. BSDi](#).



Tvrtka **BSDi** se kasnije (2000.g.) spojila s tvrtkom [Walnut Creek CDROM](#), koja je od ranih 90-tih između ostalih aktivnosti, bila distributer *Slackware Linuxa* te drugih *open source* i *freeware* programa na CD-ROMu. 2001. godine, **BSDi** prodaje svoj odjel za razvoj softvera, tvrtki [Wind River Systems](#). Sada se pod njihovim krovom nalaze distribucija i

djelomičan razvoj: **BSD/OS**, **FreeBSDa** te **Slackware Linuxa**.

Wind River ubrzo prestaje s podrškom i sponzoriranjem razvoja *Slackware Linuxa*. te je odjel koji je radio distribuciju i dio razvoja *FreeBSDa*, odvojen u zasebnu tvrtku **FreeBSD Mall**, tijekom 2002. godine.

Nakon godina previranja, dolazi tvrtka [iXsystems](#), koja kupuje dio njihovog poslovanja. Od tada, do danas *iXsystems* je zadužen za podršku, distribuciju i dio razvoja **FreeBSDa**. Potom preuzimaju razvoj **FreeNAS** sustava (koji je baziran na *FreeBSDu*), i to od inačice 7, sve do danas (v.9.x), a na bazi kojega konstantno razvija svoju komercijalnu (*Enterprise*) inačicu, imena [TrueNAS](#).

Nadalje *iXsystems* nastavlja s podrškom i sponzoriranjem razvoja:

- **Slackware Linux**
- **FreeBSD**
- **FreeNAS**
- **OpenZFS**
- **TrueOS** (nekada poznatom pod nazivom *PC-BSD* ili *PCBSD*)

Već par mjeseci kasnije, nakon izdavanja Jolitzove 386/BSD verzije, formirala se aktivna grupa korisnika tog sustava te je oformila grupu imena **NetBSD** koja je nastavila s razvojem. Početkom 1993 doradili su verziju koju su nazvali **NetBSD**.

Njihov plan razvoja je bio usmjeren na multiplatformnost, dakle mogućnost pokretanje **NetBSDa** na što većem broju hardverskih platformi. Do 1998 njihova distribucija je bila isključivo dostupna preko mreže (interneta), ali bez izrađenog instalacijskog medija (CD i sl.) te je bila namjenjena samo "hard-core" tehničkim korisnicima.

S druge strane, nekako s vremenom formiranja **NetBSD** grupe formirana je i **FreeBSD** grupa, čiji cilj je bio samo podrška za PC platformu (arhitekturu računala), s jednostavnom instalacijom te su izdali instalacijski CD-ROM disk 1993 godine.

S obzirom na jednostavnu instalaciju te veliku promociju na sajmu **Comdex** **FreeBSD** je najšire prihvaćen te se njegov razvoj drastično ubrzao.

FreeBSD je dobio dodatnu popularnost uvođenjem emulacije *Linuxa*, koja je omogućavala pokretanje *Linux* programa (binarnih datoteka) direktno na **FreeBSD** sustavu uz sve prednosti **FreeBSDa** kao pouzdanog, robusnog i performantnog operativnog sustava.

1995 godine pojavio se i **OpenBSD**, koji je izrastao iz **NetBSD** grupe. Njihov fokus je na povećanju sigurnosti cijelog sustava. Vrlo brzo su i oni krenuli s izdavanjem (i prodajom) CD-ROM instalacijskih medija.

Tehnologije koje nam je donio BSD

Pogledajmo nekoliko ključnih tehnologija koje nam je donio BSD:

Berkeley sockets

BSD Unix je bio prva inačica Unixa koja je uključivala biblioteke koje su podržavale IP grupu protokola (Engl. Internet Protocol stack). Povijesno gledano prva implementacija mrežnog sloja je odrađena od strane sveučilišta u [Illinoisu](#) 1975. godine ali to još nije bila TCP/IP grupa protokola, već preteča TCP protokola (transportni dio protokola), koja se zvala [NCP](#). S obzirom na činjenicu što je hardver u to vrijeme bio vrlo ograničen (PDP-11), a i sama implementacija je bila komplicirana rezultat su bile vrlo loše performanse. Ova inačica je srećom bila vrlo brzo dostupna na sveučilištu Berkeley gdje se nastavila razvijati (pogotovo dolaskom novog hardvera).

Godinama kasnije, s verzijom 4.2BSD (1983.g.), kako se razvijao TCP/IP, "Berkeley Socketi" su integrirani u BSD Unix.

"Berkeley socketi" su implementirani kroz niz API-ja (Application Programming Interface) i biblioteka u pozadini a koji se u konačnici koriste za komunikaciju između Unix procesa preko IPC (Engl. Inter-Process Communication) mehanizama.

"Socketi" su apstraktna reprezentacija lokalnih krajnjih točaka u mrežnoj komunikaciji. Oni su implementirani pomoću "File handle"-ova. Drugim riječima svaki pojedini "Network Socket" je krajnja točka u mrežnoj komunikaciji - za primanje i slanje podataka preko mreže, a koja se predstavlja kao posebna datoteka u Unixu (i Linuxu).

Ovom integracijom "Socket"-a u Unix operativni sustav kao *File Descriptors* tj. u konačnici kao posebne datoteke s kojih se može čitati ili u njih zapisivati, sve operacije rada s mrežom su drastično pojednostavljene i ubrzane.



Za više informacija o **File deskriptorima** pogledajte knjigu *Uvod u Linux i Linux napredno*, poglavlje [File descriptors](#) te primjere.

Binarna kompatibilnost

BSD operativni sustavi su u mogućnosti izvršavati "native" softver (onaj razvijen za druge operativne sustave) iste arhitekture za koju je napravljen (x86, MIPS, ARM i sl.). Ova nevjerojatna mogućnost postignuta je upotrebom tzv. *binary compatibility* sloja unutra operativnog sustava, koja je znatno jednostavnija i brža od bilo koje [emulacije](#). Ova funkcionalnost daje nam sposobnost izvršavanja aplikacija koje su izvorno razvijene i kompilirane za Linux i to punom brzinom i bez nepotrebnih slojeva i slojeva emulacije.

Time se proširila upotreba BSDa (pr. FreeBSDa) jer je kako smo rekli u njemu moguće izvršavati i programe za Linux, kojih možda uopće nema za BSD na kojemu radimo.

Drugi nasljednici BSDa

Pogledajmo i razne distribucije koje su nastale na :

FreeBSD distribuciji:

- [NeXT](#) i [OpenStep](#), bazirani na [Mach kernelu](#) i "4BSD" - oni su preci [macOS](#)-a
 - Apple-ov [Darwin](#), koji je jezgra [macOS](#) i [iOS](#)-a; a izgrađen je na bazi [XNU kernel-a](#) koji je opet dio "Mach" kernela, a dio FreeBSD-a te ostatka operativnog sustava koji je većim dijelom FreeBSD.
- [TrueOS](#), [GhostBSD](#) i [DesktopBSD](#), koji su bazirani na FreeBSDu
- [MidnightBSD](#), još jedan fork FreeBSDa
- [DragonFly BSD](#), također fork FreeBSDa.
- [NextBSD](#), nova BSD distribucija koja je nastala kombinacijom **FreeBSD 10.1** i raznih [macOS](#) dijelova.
- [FreeNAS](#) baziran na FreeBSD, minimalistički i optimiziran za NAS/SAN rješenja.
 - [NAS4Free](#) fork verzije 0.7 FreeNASa.
- [Nokia IPSO](#) baziran na FreeBSD a koristi se za Firewall, znane kao "Check Point IPSO".
 - Sada u vlasništvu tvrtke [Check Point](#) i osnova su njihovih Firewall rješenja
- OS za ["Netflix Open Connect"](#) uređaje
- [Junos](#), operativni sustav za Juniper Networks uređaje. Baziran na modificiranoj verziji FreeBSDa i drugih komponenti.
- Isilon Systems [OneFS](#), operativni sustav "IQ" serije klasterskih "Storage" sustava, baziran na FreeBSDu.

- [NetApp Data ONTAP](#), operativni sustav za NetApp "Storage" sustave, baziran na FreeBSDu.
- [m0n0wall](#), FreeBSD distribucija optimizirana za rad kao Firewall za "Embedded" platforme.
 - [pfSense](#), razvio se iz projekta "m0n0wall" a baziran je na FreeBSD, kao platforma za Firewall/Router/VPN uređaje.
 - [OPNsense](#) koji je nastao kao fork "pfSense" i "m0n0wall" projekata tijekom 2014.g.
- Prema nekim izvorima i dijelovi [Cisco IOS-a](#), koji je univerzalan operativni sustav za Cisco Routere i Switcheve je baziran na FreeBSDu.
- [BSD Router Project](#) - distribucija za mikro Firewall/VPN/Router uređaje.
- Sony Playstation 3 i [Playstation 4 \(Orbis OS\)](#) su bazirani također na FreeBSDu.
- [Coyote Point Systems](#) visoko performantni operativni sustav **EQ/OS**, za Load Balancing, sada u vlasništvu tvrtke **Fortinet**, u liniji proizvoda "[FortiADC Application Delivery Controller \(ADC\)](#)"

NetBSD distribuciji:

- [OpenBSD](#), fork NetBSDa, s naglaskom na sigurnosti.
- [MirOS](#), fork OpenBSDa.
- [MicroBSD](#), OpenBSD baziran sustav, vrlo male veličine.
- Force10 [FTOS](#), operativni sustav za Switcheve, tvrtke **Force 10**, sada vlasništvo tvrtke **Dell**
 - [Dell DNOS](#) od verzije 9 nasljednih FTOSa.

Koje još tehnologije imamo na FreeBSD/NetBSD/OpenBSD Unixima

Kako su FreeBSD, NetBSD, OpenBSD i drugi nasljednici BSD Unixa iz ove grane rasli i razvijali se, razvijale su se i nove tehnologije. Cijelim putem razvoja ove grane BSDa, naglasak je bio na robusnosti i izdržljivosti, kao i sigurnosti. Stoga su ugrađeni razni zaštitni mehanizmi, koji osiguravaju nesmetani rad, čak i u situacijama, u kojim bi drugi operativni sustavi imali velikih problema u radu.

Tako je primjerice FreeBSD otporan na probleme uzrokovane Tzv. **Interrupt Throttling** znan i kao [Interrupt Storm](#). Naime ovo je pojava tijekom koje procesor (CPU) dobiva veliki broj signala prekida (Engl. Interrupts odnosno IRQ) koji mu zauzimaju veliki dio vremena.

U takvim stanjima se cijeli operacijski sistem može "zaglaviti" jer je procesor preopterećen obradom signala prekida od problematičnog uređaja - odnosno onog uređaja koji je uzrokovao **Interrupt Storm**. Ovo inače ne mora nužno biti greška u hardveru uređaja koji generira ovaj "problem", već je možda taj uređaj preopterećen (ili vršno opterećen duže vrijeme).

Primjerice to može biti mrežna kartica koja obrađuje (pre) veliki broj paketa a nema dobro (ili uopće) implementiranu mogućnost ograničavanja količine signala prekida koje generira (Engl. interrupt rate limiting).



Većina današnjih operativnih sustava se potpuno i bezizlazno zaglavi u ovakvom slučaju. U Windows operativnom sustavu je ovo poznati *Blue Screen of Death* u kojemu je jedino rješenje hardversko *resetiranje* računala.

Dok primjerice FreeBSD detektira ovakav slučaj, te maskira problematični signal prekida (*interrupt*) na neko vrijeme, te ga ponovno nastavlja obrađivati (uz dodatne mehanizme), bez problema u radu ostatka sustava. Zanimljivo je i da se Linux nepovratno zamrzava u ovakvim slučajevima.

Dodatno, OpenBSD zajednica, je zadužena za razvoj i održavanje mnogih tehnologija i protokola, uz znatno sigurniju implementaciju. Njihova implementacija se opet, koristi i na FreeBSD/NetBSD i drugim operacijskim sustavima. Pogledajmo neke od tih tehnologija i protokola :

- [LibreSSL](#) - fork *OpenSSL*a, koji implementira **SSL** i **TLS** protokole.
- [OpenSSH](#) - open source implementacije **SSH** protokola
- [PF](#) (*Packet Filter*) koji je IPv4/IPv6 *stateful firewall* s NAT, PAT, QoS i naprednim mogućnostima baratanja s mrežnim prometom. PF se koristi i primjerice u FreeBSD Unixu.
 - [pfsync](#), alat/funkcionalnost koja omogućava sinkronizaciju stanja (Engl. States) *Firewalla* (sinkronizaciju svih *Sessiona*) između dva ili više PF uređaja/poslužitelja/*Firewalla*. Dodatno ima podršku za sustav visoke dostupnosti, upotrebom [CARP](#) protokola. CARP se koristi za mrežnu redundanciju između dva ili više mrežnih uređaja ili računala, vrlo slično kao [VRRP](#) protokol. *pfsync* i CARP se također koriste i na FreeBSD Unixu.
- [CARP](#) (Common Address Redundancy Protocol) - protokol za redundanciju - alternativa za [Cisco HSRP](#) ili [VRRP](#) protokole.
- [OpenBGPd](#) je open source implementacija *Border Gateway Protocol 4* (BGP-4) routing protokola.
- [OpenOSPFd](#) je open source implementacija *Open Shortest Path First* (OSPF) routing protokola.
- [OpenNTPD](#) je open source implementacija *ntp.org Network Time Protocol* (NTP) servisa/*daemon*a
- [OpenSMTPd](#) je open source implementacija *Simple Mail Transfer Protocol* (SMTP) s podrškom za IPv4 i IPv6.
- ... uz još mnoge druge.

Ne zaboravimo niti **ZFS**, čiji izvorni kod je prenesen sa **Sun Solaris 10** operativnog sustava, jer je kompatibilan sa BSD licencom. Dakle govorimo o najnaprednijem datotečnom sustavu, koji je da budemo detaljniji, kombinacija *Logical Volume Managera* (ekvivalent naprednom **RAID** kontroleru) i datotečnog sustava.

Priča o tvrtki Sun Microsystems

Za vrijeme studija na sveučilištu Stanford u Kaliforniji, [Andy Bechtolsheim](#) je za projekt “**Stanford University Network communications**”, napravio računalo koje je nazvao *Sun-1*. Osnovna namjena mu je trebala biti upotreba kao radne stanice za [CAD](#) programe. Ova radna stanica je koristila Motorolin procesor [68000](#) koji je imao ugrađen napredni **MMU** (Engl. Memory Management Unit) a koji je omogućavao Unixu, upotrebu sustava virtualne memorije.

Prvu verziju radne stanice **Bechtolsheim** je napravio od rezervnih dijelova koje je dobio od sveučilišnog odjela računalnih znanosti te drugih dobavljača koji su radili u Silicijskoj dolini.



Od 1940 do 1950 godine, tadašnji dekan [Frederick Terman](#), je poticao sveučilište i studente na pokretanje tehnoloških tvrtki. Njegov plan je bio iskoristiti zemljište u vlasništvu *Stanforda* i pretvoriti ga u Tzv. **Stanford Industrial Park** (kasnije preimenovan u **Stanford Research Park**).

Dodatno, osmislio je kooperativni program, od kojega su koristi imale nove tvrtke, studenti i sveučilište.

Stanford Industrial Park je tako postao prvi poslovni inkubator za buduće tehnološke tvrtke, kojima bi sveučilište pomagalo u radu i razvoju. Sve tvrtke su potpisivale ugovor na pet godina, što se smatralo dovoljnim, za samostalan dalji rad i razvoj. Jedna od prvih tvrtki koja se tamo doselila je i poznati [Hewlett-Packard](#), koju su osnovala dvojica Stanfordskih studenata (William Hewlett i David Packard). Malo po malo, broj tvrtki je rastao, te se sve pretvorilo u ono što je danas poznato kao **Silicijska dolina**.



Zanimljivo što je i nastanak tvrtke **Cisco Systems** usko vezan uz *Stanford* i zahvaljujući njemu (uz određene *kontroverze*). Za više detalja pogledajte [Wikipediju](#).

Početkom 1982 godine, radu **Bechtolsheima** su se priključila još dvojica Stanfordskih studenata: [Vinod Khosla](#) i [Scott McNealy](#) koji su u okviru Stanfordskog inkubatora pokrenuli tvrtku **Sun Microsystems**. Vrlo brzo im se priključio i Bill Joy s *Berkeleya* (poznat po razvoju BSD Unixa) tako da se i on smatra jednim od osnivača.



Ime tvrtke **Sun** je nastalo od riječi *Stanford University Network*

Već 1983. godine, krenula je proizvodnja grafičke radne stanice bazirane na Motorolinom procesoru 68000, nakon čega su uslijedile nove verzije na sve jačim Motorolinim procesorima:

- **Sun-2** - Motorola 68010
- **Sun-3** - Motorola 68020
- **Sun-3 X** - Motorola 68030

SPARC procesor(i)

Negdje 1987 godine tvrtka je razvila vlastiti procesor (*CPU*), imena [SPARC](#), [RISC](#) arhitekture. Uslijedile su nove generacije računala (počevši od **Sun-4**), na ovom procesoru. Prvi SPARC procesori su bili 32. bitni (SPARC V7 arhitektura), do razvoja 64.bitnih SPARC procesora 1995 godine (SPARC V9 arhitektura).

Početkom 1990-tih krenuli su u razvoj SPARC procesora s više jezgri, Tzv. [SMP](#) (Symmetric Multi Processing) procesori. Njihov naziv je bio [UltraSPARC](#). Samim time se proširila i namjena, sada se pojavljivao sve veći broj *Sunovih* poslužitelja. S vremenom se sve više povećavao broj jezgri i funkcionalnosti te su uslijedile nove generacije ovih procesora:

- **UltraSPARC II** , **UltraSPARC III** , **UltraSPARC IV** , **UltraSPARC IV+** te
- **UltraSPARC V** (koji je otkazan)

Potom je uslijedio prelazak na drugi dizajn, pa su se pojavile nove generacije:

- [UltraSPARC T1](#) (2005.g.) - imao je 8 jezgri sa po 4 niti na svakoj jezgri, što čini ukupno **32** logičke jezgre. Jedina slabost ovog dizajna je bila u tome što se jedna **FPU** (Engl. *Floating Point Unit*) (matematički koprocessor) jedinica dijelila između svih osam jezgri, pa su određene operacije bile koju su zahtijevala **FPU**, u višenitnom radu, bile sporije.

- [UltraSPARC T2](#) (2007.g) - imao je 8 jezgri sa po 8 niti na svakoj jezgri, što čini ukupno **64** logičke jezgre, uz integrirane dvije 10Gbps "mrežne kartice" na samom procesoru. Ovaj dizajn je bio značajan napredak u odnosu na prethodnu inačicu (*T1*) jer se sada za svaku fizičku jezgru (njih osam) dodao po jedan **FPU**, dakle sada ih je bilo osam. Dodane su i dvije **ALU** jedinice za rad s cijelim brojevima (Engl. **Integer**), po jezgri. Promijenio se i dizajn i veličina **cache** memorija te **cjevovodi** i drugo. Ovim potezima su se drastično ubrzale matematičke i operacije kod višenitnog rada.
- Zatim su uslijedili : **UltraSPARC T2+** , **UltraSPARC T3** te **UltraSPARC T4** i
- [UltraSPARC T5](#) - koji je imao 16 jezgri te po 8 niti na svakoj jezgri, što čini ukupno **128** logičkih jezgri



Sun Microsystems je i prije razvoja **UltraSPARC T1** arhitekture procesora, patentirao tehnologiju u kojoj se na jednoj fizičkoj jezgri može izvršavati više **threadova** (niti) ali ju je ozbiljnije počeo upotrebljavati upravo na ovoj generaciji procesora. Te je već u slijedećoj generaciji **UltraSPARC T2**, koja je na tržište izašla 2007.godine, došao do brojke osam (8). Dakle na jednoj fizičkoj jezgri se može pokretati do osam (8) niti, što sustav vidi kao osam logičkih jezgri a što je drastično ubrzalo rad sustava i aplikacija.



Isti koncept je kasnije upotrijebio i **Intel**, koji ju je nazvao **Hyper-Threading**, uz sličan dizajn poput **UltraSPARC T2**, barem što se tiče **FPU**, **Integer ALU** jedinica, **cjevovoda** i **cache** memorije. Stoga *Intel*ova implementacija nije imala bolje slabijih performansi kod upotrebe višenitnog rada, odnosno kod iskorištavanja svih logičkih jezgri. Iako se **Intel** nikada nije niti približio broju od osam (8) niti po fizičkoj jezgri - niti danas (2017.g.).

S druge strane **AMD** je malo kasnije ušao u ovakav dizajn, te su svi njegovi procesori patili od iste boljke koju je imao **UltraSPARC T1**, proizveden 2005. godine. Naime tek razvojem **AMD Zen** arhitekture, odnosno prvih procesora kodnog imena **Ryzen**, koji će izaći u trećem mjesecu 2017. godine, **AMD** bi prema specifikaciji, trebao imati mogućnost izvršavanja više niti na jednoj fizičkoj jezgri uz dovoljan broj **FPU** i **Integer ALU** jedinica i ostalih komponenti, koje su potrebne za brz višenitni rad.

Tranzicija u Open Source

I konačno, uslijedio je prelazak u open source svijet. Naime do tada je *Sun* licencirao i prodavao licence za procesore SPARC, drugim proizvođačima, koji su ih proizvodili.

2006. godine odlučeno je da će se dizajn **UltraSPARC T1** procesora objaviti prema novoj licenci : GNU GPL. Dakle sve od dizajna i simulatora, do cijele arhitekture **Ultra SPARC** je objavljeno u GNU GPL licenci. Stoga je pokrenut projekt imena [OpenSPARC](#).

2007. godine *Sun* je objavio i cijeli dizajn procesora **UltraSPARC T2** u GNU GPL licenci, te je iz njega nastao **OpenSPARC T2** s 8 jezgri, 16 cjevovoda sa 64 niti (threads) odnosno **64** logičke CPU jezgre.

x86 i x64 procesori

Sun je istovremeno s radom na SPARC procesorima, radio i radne stanice a kasnije poslužitelje sa **Intel** i **AMD** x86 (32.bitni) i x64 (64.bitni) procesorima.

SunOS, Solaris i napredne tehnologije

Operacijski sustav **SunOS**, razvijen je na osnovi *BSD Unixa* za prva računala tvrtke *Sun Microsystems*. Razvoj se nastavio sve do inačice 4 (*SunOS v.4*) ali se tada ime operacijskog sustava promijenilo u **Solaris**.

Solaris je ubrzo sa *BSD Unixa* kao baze, prešao na **System V Release 4**, već s inačicom **Solaris 2** (1992. godine). Kupnjom raznih tvrtki i njihovih patenata, i dolaskom do njihovih tehnologija, dodatno se razvijao i *Solaris*, kao i razni sustavi i programi koji su uslijedili.

Od *SunOSa* do *Solarisa*, razvijale su se i razne tehnologije i protokoli, koji su i danas standardi, a neki od njih su superiorni i danas. S vremenom je tvrtka **Sun Microsystems**, kupovala, ne samo rješenja, već cijele tvrtke i njihove patente. Neke od tvrtki i tehnologija koje su kupili su i:

- [CRAY](#) - tvrtka koja je proizvodila superračunala - s njom su dobili cijeli niz tehnologija (1996.godine)
- **Integrated Micro Products** - tvrtku specijaliziranu za poslužitelje otporne na ispade (Engl. Fault tolerant servers) (1996.godine)
- [Encore Computer](#) - tvrtku specijaliziranu za "Parallel computing" dakle sustave koji su bili sastavljeni od velikog broja prosječnih "računala" ali su se ponašali kao jedno veliko superračunalo (1997.godine) - ovo je tada bila vrlo napredna tehnologija
- **Dakota Scientific Software** - tvrtku koja je razvila razvojne alate za razvoj *high-performance computinga* - također vrlo važne za superračunala i druge primjene (1998.godine)
- **StarDivision** - tvrtku koja je razvila *StarOffice* uredski paket softvera - koji je kasnije objavljen po open source licencom - kao *OpenOffice* (1999.godine).
- **MAXSTRAT** - tvrtka koja razvija *Fibre Channel storage* poslužitelje - koja drži važne tehnologije za *Storage* sustave. (1999. godine)
- **Gridware** - tvrtka čija softverska rješenja su zadužena za distribuirani rad (raspoređivanje poslova na više poslužitelja unutar *GRID* okruženja). *GRID* se smatra skupom više *clustera*, a *cluster* skupom velikog broja računala. (2000. godine)
- **LSC** - tvrtka koje je razvila *Storage and Archive Management File System (SAM-FS)* te *Quick File System (QFS)* - datotečne sustave za izradu sigurnosnih kopija (*backup*) i arhiviranje. (2001.godine)
- [StorageTek](#) - tvrtku specijaliziranu za pohranu podataka (Engl. *Data storage*) - razvijali su tehnologije i proizvode, od polja diskova, tračnih uređaja i spremišta, do SCSI i SAS RAID kontrolera i sl. (2005. godine)
- [MySQL AB](#) - tvrtka koja je razvila **MySQL** bazu podataka, koju je Sun Microsystems i nastavio razvijati kao open source (2008. godine)
- ... deseci drugih

Sun Microsystems je u trenutku objavljivanja inačice **Solaris 10** operacijskog sustava, odlučio otvoriti kompletan programski kod, pod otvorenom CDDL (Common Development and Distribution License) licencom, tijekom 2005. godine.



Već 2006 godine cijeli programski kod *Solarisa 10* prebačen u projekt koji je *Sun Microsystems* pokrenuo, a koji je nazvan [OpenSolaris](#).

Inačica 10 **Solarisa** je mnogim područjima bila tehnološki superiorna gotovo svim operacijskim sustavim toga vremena, te je uključivala cijeli niz tehnologija koje su tada postale javno dostupne. Tako je bilo sve do 2010. godine kad ih je kupila tvrtka **Oracle** koja je već u slijedećoj inačici (**Solaris 11 Express**) zatvorila programski kod.



Ovim potezom se nije ništa posebno promijenilo jer je cijeli otvoreni kod (do tog trenutka) preuzet, od strane pojedinaca i grupa ljudi, te kasnije raznih organizacija i tvrtki, koje su ga nastavile razvijati kao otvoreni kod. Iz tog koda su sve napredne tehnologije ubrzo postale dostupne i za druge operacijske sustave, a koji su kaskali desetljeće iza njih (**govorimo o određenim tehnologijama**).

Neke od tih naprednih tehnologija su:

- [DTRACE](#) - *dynamic tracing framework* za praćenje rada i *troubleshooting* od razine kernela do aplikacije, u realnom vremenu. Omogućava vrlo detaljan pregled i analizu rada bilo koje komponente sustava.
- [ZFS](#) (Zettabyte File System) - najnapredniji datotečni sustav svih vremena, koji je i *Logical Volume Manager* (ekvivalent naprednom RAID kontroleru) i napredni datotečni sustav - sve u jednom. Razvijen kao izvrsna zamjena za RAID kontrolere i tadašnje datotečne sustave.
- [Solaris Containers](#) - tehnologija "kontejnera" unutar operacijskog sustava, koja omogućava "virtualizaciju" (koja nije virtualizacija) na razini operativnog sustava
- Mrežni model i kernel s naprednim funkcionalnostima



Sun Microsystems je u svoje vrijeme (prije kupnje od tvrtke **Oracle**) javno objavio (u nekoj od *open source* licenci) cijeli niz tehnologija, programa, i alata :

- Programski jezik [JAVA](#) - 2005 objavljen pod GPL licencom (sadržavao je 6.5 milijuna linija programskog koda)
- [OpenOffice](#) uredski paket programa - objavljen pod LGPL i SISSL licencama (sadržavao je 10 milijuna linija programskog koda)
- [GlassFish](#) Java aplikacijski poslužitelj - objavljen pod CDDL i GNU GPL licencama
- [MySQL](#) bazu podataka i pripadajuće alate
- [VirtualBox](#) softver za virtualizaciju - objavljen pod CDDL, GNU GPL i drugim licencama
- već navedene (*ZFS*, *Solaris containers*, *DTRACE*) i mnoge druge
- druge



Sun Microsystems je u svojim zlatnim vremenima slovio kao inženjerska tvrtka, u kojoj su inženjeri razvijali programe i sustave za inženjere - na najbolji mogući način, kao nedostižni uzor svim ostalim tvrtkama (barem za 99.9999% njih). Što se hardvera tiče, gdje god se nešto moglo predimenzionirati i napraviti kvalitetnije i pouzdanije, da bi trajalo duže, to se i uradilo. Što se softvera tiče, sve se razvijalo na najbolji mogući način, prema svim pravilima struke.

Druge paralelne priče

Još davno prije odluke u razvoju BSD Unixa, u trenutku kada je odlučeno da će se svi programi za BSD Unix početi pisati (razvijati) od početka, pod BSD licencom, s otvorenim programskim kodom, dogodilo se nešto slično.

Ponovno se vraćamo u prošlost

GNU i Linux

Od kraja 1970.g. Unix je sve više bio prihvaćan, od državnih institucija i sveučilišta do većih tvrtki. Pošto je još uvijek bio vezan za skupi hardver a i cijena licenci je polagano rasla, nije bio prihvaćen u malim tvrtkama ili od strane pojedinaca. U tom trenutku, cijeli programski kod - od samog Unixa do pratećih programa, bio je javno dostupan (Tzv. Open Source).



To što je programski kod bio objavljen i dostupan, ne znači da se nisu morale plaćati licence !.

Kako su troškovi razvoja rasli, a vjerojatno je rasla i pohlepa za zaradom, polagano je razvoj krenuo u smjeru zatvaranja programskog koda za javnost. Dakle više nitko osim vlasnika takvih programa, više nije imao pristup programskom kodu, koji bi mogao optimizirati, proširiti ili bilo kako mijenjati. Dodatno pojavili su se novi zakoni koji su autorska prava drastično proširili i na ovo područje.

Tih godina [Richard Stallman](#) je bio student na sveučilištu MIT, te član hakerske zajednice unutar koje se programski kod slobodno dijelio. Nedugo zatim, trend zatvaranja koda se proširio te pogodio i hakersku zajednicu, kojoj je Stallman bio član. Frustriran ovim promjenama, [Richard Stallman](#) pokreće 1983. godine **Free Software Movement** inicijativu za razvoj aplikacija otvorenog koda. Već 1984. godine pokreće i [GNU projekt](#), koji kreće s razvojem prvih programa otvorenog koda (Open Source).

1985 godine, Stallman objavljuje dokument [GNU Manifesto](#), u kojemu opisuje problematiku i uzroke pokretanja GNU projekta, kao i daljnje ciljeve u radu. Ovim dokumentom želio je senzibilizirati zajednicu na negativne trendove u razvoju (zatvaranje programskog koda), te dati svoj doprinos i s moralne i filozofske strane.



Jedan od važnijih krajnjih ciljeva GNU projekta je razvoj slobodnog operativnog sustava imena **GNU**. Put do toga je trebao proteći u razvoju svih programa koji bi bili potrebni za operativni sustav. Pod pojmom "slobodnog" se misli na: "slobodnog za korištenje, mijenjanje i distribuciju".

Iste godine [Richard Stallman](#) osniva i zakladu [Free Software Foundation \(FSF\)](#) kojoj je inicijalni cilj podupiranje razvoja "Free Software Movement" inicijative, koja promovira slobodu učenja, distribucije, stvaranja i modificiranja računalnih programa.

S ovim korakom jača se i pravna potpora svim navedenim projektima i u konačnici razvoju programa i sustava otvorenog koda.

Prema načelima **FSF** nastala je prva od licenci otvorenog koda Tzv. **GNU GPL** odnosno *GNU General Public License* ili prevedeno *GNU opća javna licenca*, ali tek kasnije, negdje tijekom 1989. godine. Naime sve do tog trenutka licence većine programa oduzimale su vam mogućnost (slobodu) dijeljenja s drugima, mijenjanja njenog izvornog koda (ako je uopće bio dostupan) i distribucije istog. Druge licence su vam to sve omogućavale ali uz znatna financijska sredstva, koja su rasla iz godine u godinu.



Kasnije se s ovakvim sličnim (**FSF**) načelima razvijao i BSD Unix. Sjetite se poglavlja ["Tranzicija BSD UNIXa u Open Source"](#) (1989.g.). Dakle ove dvije priče iako razdvojene, krenule su s istom željom.



GNU GPL licenca je Tzv. **Copyleft** licenca, koja govori kako programski kod koji je napisan, može biti kopiran ili mijenjan. Ali sav programski kod nastao od izvornog koda također mora biti objavljen pod istom licencom dakle javno dostupan. To znači da ako uzmete programski kod objavljen pod GNU GPL licencom i mijenjate ga ili proširujete, sve što ste radili MORATE javno objaviti i to ponovno pod istom licencom. Dodatno to znači da ne smijete razvijati svoje programe ili sustave bazirane na GNU GPL programskom kodu ako ih ne mislite dijeliti s drugima, odnosno ako ih želite sakrivati ili zatvarati od drugih.

Ponovni povratak u povijest !

Vratimo se na 1984.godinu i GNU projekt

Želja je bila sve programe koji postoje u komercijalnim Unix operativnim sustavima napisati “od nule” odnosno ne koristeći niti jednu liniju koda od izvornih programa, jer bi to bilo kršenje autorskih prava. Dakle željelo se napisati nove programe s istom ili proširenom funkcionalnosti, postojećih i objaviti ih pod nekom od Open Source Licenci.



Veći dio ovog posla (ovdje govorimo o stotinama i stotinama programa) je dovršen negdje u toku 1992. godine. Cijeli niz programa koje su razvijali, postaje poznat pod nazivom *GNU paketi* ili *GNU programi*.

Slika : Richard Stallman (2014.g).



[Richard Stallman](#) zaslužan je za razvoj:

- [Free Software Foundation](#) zaklade
- [GNU projekta](#)
- [Emacsa - napredni tekst editor](#)
- [GCC \(GNU Compiler Collection\) - znan kao : gcc compiler](#)
- [GNU Debuggera](#)
- [GNU make - Build automation alata.](#)

Neovisno o razvoju **FSF** i **GNU** 1991. godine Linus Torvalds razvija jezgru (Kernel) operativnog sustava i slijedi logičan korak tj. povezivanje stotina i stotina programa pisanih unutar GNU pokreta i **Linus**ovog kernela.

Od tog trenutka povezano je sve što je bilo potrebno za stvaranje novog operativnog sustava danas poznatog pod nazivom Linux (GNU Linux).

Autor: By Thesupermat - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=37264906>



U to vrijeme, pojavile su se i neke kontroverze oko naziva novog operativnog sustava nazvanog **Linux**. Naime Stallman i FSF zaklada nisu priznavali naziv Linux kao naziv za cijeli operativni sustav. Oni su inzistirali na nazivu **GNU/Linux** ili **GNU+Linux** stoga što su tražili neko priznanje za dugogodišnji rad tisuće programera, koji su razvijali one stotine i stotine GNU programa, koji su sada praktično zalijepljeni uz Linux kernel, te postali dio operativnog sustava, koji je zaboravio na njih (barem u imenu).



Sve današnje distribucije Linuxa koriste i GNU programe i programe razvijene pod BSD licencom, tako da smo dobili znatno veći broj korisnih programa. Osim većeg broja programa, dogodilo se i to da su se pojavili programi s istom funkcionalnosti, jedan nastao i razvijen iz GNU projekta a drugi iz BSD priče. Ovo se pokazalo vrlo dobrim jer se sada pojavila mogućnost odabira: boljeg, bržeg i sigurnijeg programa.

Pojavili su se i deseci drugih licenci, te stotine programa koji su razvijani pod njima. Na kraju, broj dostupnih programa se drastično povećao, na tisuće i tisuće, pa tako prosječna instalacija bilo koje od distribucija Linuxa, dolazi s nekoliko tisuća instaliranih programa.

Kako je uopće došlo do razvoja Linuxa

[Linus Benedict Torvalds](#) je tijekom studija, na Helsinškom sveučilištu, 1990. godine prvi puta susreo UNIX i to na **DEC MicroVAX** poslužitelju, koji je pokretao **ULTRIX** (DEC-ova varijanta Unixa). Ovaj, za njega novi operativni sustav bio je toliko poticajan, da je za diplomski rad odabrao temu, naziva (Linux) **A Portable Operating System** odnosno prijenosni operativni sustav.

Torvalds je kasnije radio i na MINIXu, koji je jedna od varijanti UNIXa, namijenjena uglavnom akademskim zajednicama. Kako je razvoj **kernela** radio na MINIXu, u rad je uključio i neke od programa iz njega. Konačni cilj mu je bio pokrenuti svoj operativni sustav na novom **386 PC** računalu koje je kupio.



MINIX je napravio [Andrew S. Tanenbaum](#), profesor računalnih znanosti. Objavio ga je 1987. godine kao minimalni UNIXoidni operativni sustav, odličan za učenje o principima rada operativnih sustava. Cijeli izvorni kod MINIXa je objavljen ali pod licencom koja nije bila otvorena kao recimo GNU GPL. Tek negdje u toku 2000. godine, licenca je promijenjena u BSD licencu, koja je znatno otvorenija.

Profesor **Andrew S. Tanenbaum** je objavio i knjigu o MINIXu (*Operating Systems: Design and Implementation*), koja je također Linusu bila inspiracija za razvoj svog operativnog sustava.

Zanimljivo da je za razvoj kernela koristio **GNU C compiler**, razvijen u GNU projektu, iako je bilo i drugih dostupnih **C** kompilera.



GNU C Compiler se i danas najčešće koristi za razvoj Linux kernela i mnogih drugih sistemskih komponenti za Linux.

25.8.1991 godine, Linus je na [Usenet](#) news grupi : "**comp.os.minix**" objavio:

Hello everybody out there using minix - I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

Kasnije, kako se kernel razvijao, dodavao je i druge komponente i programe, kako bi mogao izgraditi rudimentarni operativni sustav. Većina tih programa, preuzeta je iz MINIXa. Kako se novi operativni sustav **Linux** razvijao, svi MINIX programi su izbačeni i na njihovo mjesto su došli **GNU** programi. I sam Linux kernel je kasnije objavljen pod **GNU GPL** licencom.



Linus Torvalds je zaslužan za :

- [Linux kernel](#)
- [Linux](#)
- [GIT](#) sustav za verzioniranje



Linus Torvalds je, gledajući nazad, u vrijeme razvoja Linuxa, izjavio

1. Da je **GNU kernel** bio dostupan 1991. godine, on vjerojatno ne bi radio svoj kernel (**Linux**)
2. Da razvoj **386BSD** (iz kojeg je nastao **FreeBSD**, **NetBSD** i dr.), nije kasnio, zbog pravnih problema, odnosno da se pojavio 1991.g., on vjerojatno također ne bi radio na Linuxu.

Slika : Linus Torvalds (2002.g):

Autor: By Unknown photographer who sold rights to the picture to linuxmag.com - Linuxmag.com; The image is from an article in a December 2002 issue of Linux Magazine[1], CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=17991>

Od tada do danas pojavile su se razne varijante Linux-a, od kojih su se ustalile tri grane na osnovi kojih su nastali na deseci drugih Linux distribucija:

- Debian
- Slackware
- RedHat

Iz navedenih distribucija, nastale su na stotine i stotine drugih distribucija, od kojih je veliki broj i danas u upotrebi. Čak je došlo do toga da je na osnovi navedenih distribucija nastalo nekoliko većih distribucija, na kojima su se razvile nove distribucije.

Tako je primjerice na bazi *Debian*a, razvijen *Ubuntu*, a samo na bazi njega postoji **69** raznih distribucija Linuxa.

Linux se danas intenzivno koristi na raznim poslužiteljima i stolnim računalima. Osim toga, koristi se i na velikom broju specijaliziranih uređaja:

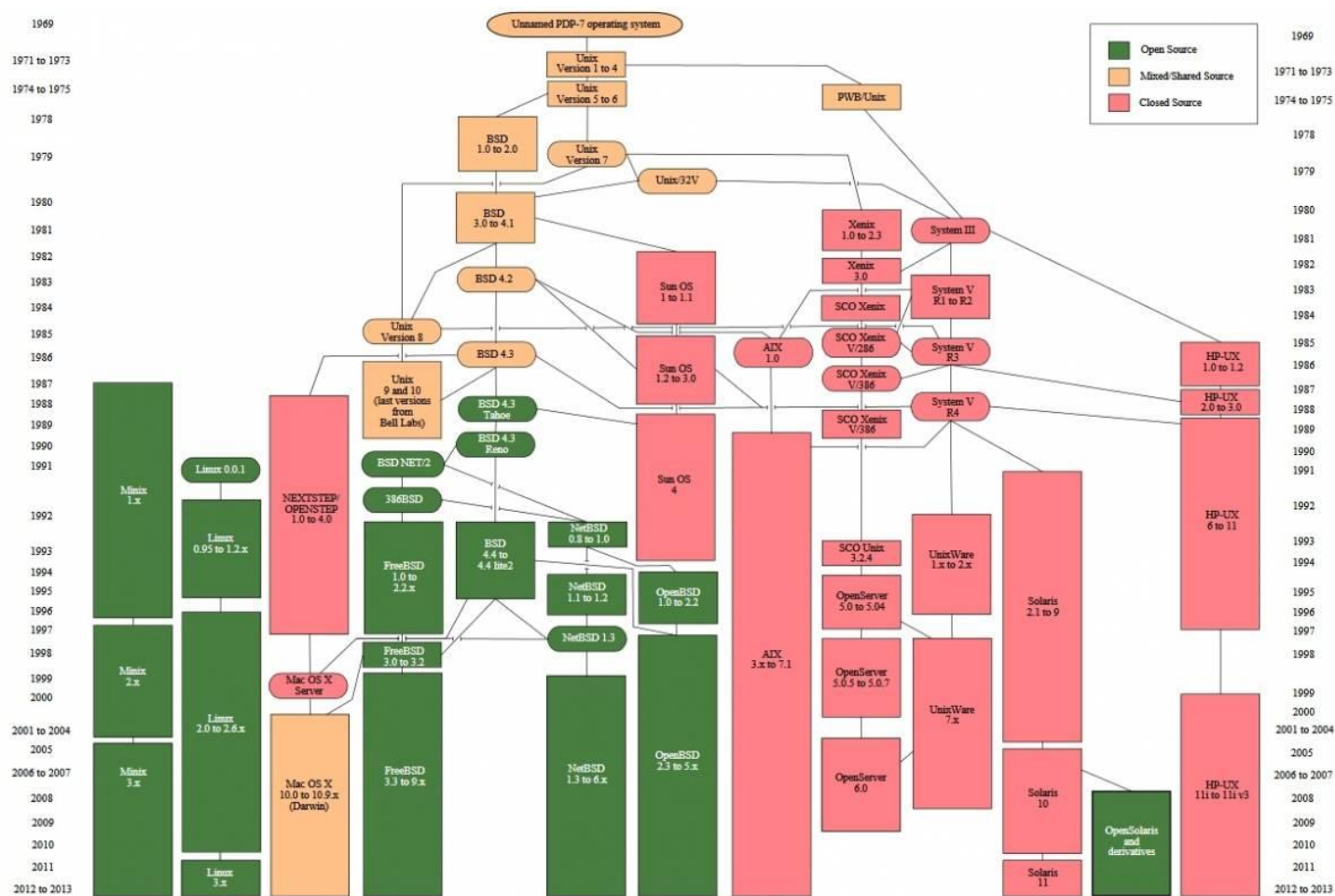
- set-top box i drugi multimedijalni uređaji
- uređaji za navigaciju
- mrežni uređaji : switchevi (preklopnici), routeri (usmjerivači) i sl.
- kućanski aparati

Nadalje, koristi se u mnogim državnim institucijama, akademskoj zajednici (škole, sveučilišta, fakulteti) i industriji. 99.6% najjačih poslužitelja na svijetu koristi Linux (Pogledajte [TOP 500](#) - popis 500 najjačih superračunala na svijetu).

Pogledajte knjigu “*Uvod u Linux i Linux napredno*”, poglavlje “[Tko sve koristi Linux i sustave otvorenog koda](#)”

Od početka do danas

Na slici je vidljiv razvoj raznih inačica Unix operativnih sustava od 1970 godine do danas.



Izvor slike : https://en.wikipedia.org/wiki/History_of_Unix#/media/File:Unix_history-simple.svg

Autor : Levenez Unix History Diagram, Information on the history of IBM's AIX on ibm.com "User:Eraserhead1, Infinity0, Sav_vas"

Unix Ratovi

Naravno sve nije nikako moglo proći bez tužbi, koje su bile jedne u nizu vezanih za UNIX. One su poznate pod nazivom "UNIX ratovi". O njima nećemo puno pričati, pa sami proučite:

Pogledajte članak na Wikipediji: https://en.wikipedia.org/wiki/Unix_wars : Unix Wars

Detaljnije o AT&T tužbama protiv

BSDa: http://www.softpanorama.org/People/Torvalds/Finland_period/att_lawsuit_as_a_launcher_for_linux.shtml "AT&T lawsuit helps to launch Linux into mainstream"

Pogledajte i ovu tužbu : *SCO Group vs Novell* (vs Unix vs

Linux) https://en.wikipedia.org/wiki/SCO_Group,_Inc._v._Novell,_Inc._:SCO_Group_vs_Novell

Pogledajte i: https://en.wikipedia.org/wiki/UNIX_System_Laboratories,_Inc._v._Berkeley_Software_Design,_Inc._%22USL_vs_BSD%22 "USL vs BSD"

Rezimirana povijest Unix-a

Operativni sustav Unix razvili su znanstvenici Ken Thompson i Dennis Ritchie u tvrtki **AT&T** radeći u **Bell Laboratories** istraživačkim laboratorijima krajem 1969 godine. Prva radna inačica nastala je početkom 1970 godine. Prve verzije UNIX-a su bile razvijane u programskom jeziku "Assembler" a kasnije, negdje tijekom 1972, cijeli Unix je napisan od početka, korištenjem programskog jezika "C".

Prelaskom na programski jezik C povećala se portabilnost softvera, odnosno samog UNIX-a i na druge platforme odnosno arhitekture procesora.

Naime programski kod pisan u programskom jeziku "Assembler" je čvrsto vezan za arhitekturu procesora te ga je bilo vrlo teško prepisati za neku drugu arhitekturu. Prelazak na programski jezik C je pojednostavio ovaj proces, pri čemu je zadržana brzina rada.

Danas se Unix koristi na različitim arhitekturama procesora poput:

- **x86**
- **PowerPC**
- **mips**
- **SPARC**
- **ARM**



1.1.1970 se smatra rođenjem Unixa

Krajem 1970 te početkom 1980 tvrtka "AT&T" licencirala je Unix drugim tvrtkama, koje su krenule s razvojem svojih inačica Unixa.



UNIX operativni sustavi, su u širokoj upotrebi preko 40 godina i oni su utjecali na razvoj drugih operativnih sustava. Osim toga, veliki broj organizacija, tvrtki i pojedinaca su bili i jesu uključeni u njegov razvoj i napredak.

Neke od inačica UNIXa, koje su i danas u upotrebi su:

- **BSD Unix** (Berkeley Software Distribution) - razvijen u Kalifornijskom sveučilištu "Berkeley"
 - **FreeBSD** baziran na BSD Unixu
 - **DragonFly BSD** baziran na FreeBSDu
 - **JunOS** - baziran na FreeBSD, razvijen u tvrtci Juniper Networks
 - **NeXTSTEP** (kasnije **Mac OS/iOS**) - razvijen u tvrtki "**NeXT Computer**", kasnije kupljen od tvrtke "**Apple**"
 - **NetBSD** baziran na BSD Unixu
 - **OpenBSD** baziran na BSD Unixu
 - **TrueOS** baziran na BSD Unixu
- **SUN OS** (kasnije Solaris) - razvijen u tvrtki **Sun Microsystems**, sada u vlasništvu tvrtke **Oracle**
- **IRIX** - razvijen u tvrtki **Silicon Graphics**
- **HP-UX** - razvijen u tvrtki **HP**
- **AIX** - razvijen u tvrtki **IBM**
- **Digital UNIX** - razvijen u tvrtki **DEC** (Digital Equipment Corporation), kasnije **Compaq**, danas **HP**
- **QNX** - razvijen u tvrtki "Quantum Software Systems", kasnije "QNX Software Systems", potom "BlackBerry"
- **Minix** (Nastao od riječi "mini-Unix") - razvijen od strane "Andrew S. Tanenbaum"
- ...



Zanimljivo je da je i tvrtka **Microsoft** kupila licencu za Unix od tvrtke **AT&T**, krajem 1970, koji je razvijala pod imenom **Xenix**. Njihova inačica Unixa je sredinom 1980 postala i jedna od češće korištenih varijanti Unixa, ali ju je kasnije Microsoft prodao tvrtki "**Santa Cruz Operation (SCO)**" koja ju je nastavila razvijati pod imenom **SCO UNIX**, a danas se prodaje pod imenom **SCO OpenServer**

Detalniji razvoj UNIX-a od 1969 do danas vidljiv je u ovom dokumentu (koji se stalno osvježava) :

<http://www.levenez.com/unix/unix.pdf>

Za Unix možemo reći

- Da podržava "Multitasking" – tj. pokretanje ili izvršavanje više zadataka (programa) istovremeno,
- Da podržava višekorisnički rad – moguć je rad više korisnika istovremeno,
- De je interaktivan, što znači da je izvođenje naredbi trenutno.

Što je specifično za sve UNIXoidne operativne sustave ?

Svi Unixoidni operativni sustavi imaju zajedničko nekoliko stvari:

- koriste hijerarhijski datotečni sustav, sa definiranom i razrađenom strukturom direktorija
- konfiguracijske datoteke za sve programe i komponente sustava su "obične" tekstualne datoteke
- svi uređaji - od diskova (CD/DVD, tvrdih ili SSD), preko mrežnih, grafičkih i drugih kartica su također datoteke (doduše posebna vrsta). Stoga se komunikacija sa svim uređajima (hardverom) svodi na rad s datotekama (pisanjem ili čitanjem u ili iz njih). Čak se i komunikacija između procesa tj. **IPC** (inter-process communication) svodi na rad s datotekama.
- bilo koji program se može pokrenuti u kombinaciji s drugim programima, na način u kojemu se obrađeni podaci koje nam daje prvi program mogu proslijediti drugom programu, a ono što je on obradio, može se poslati trećem i tako dalje, stvarajući nove mogućnosti. Bez potrebe za pisanjem novog programa, već kombinirajući funkcionalnosti postojećih programa. Ova mogućnost se zove "Pipe" ili cijev za komunikaciju te izgleda ovako :

program 1 | program 2 | program 3

Osnovna filozofija Unix-a kaže : "Everything is a file" – sve je datoteka i prema tome baratanje sa samim operativnim sustavom, njegovim aplikacijama i njegovim hardverom je vrlo jednostavno. Nadalje moguće je skriptirati i automatizirati svaki djelić sustava, prema potrebi, upotrebom **shell** skripti.

Izvori informacija

[Wikipedia : UNIX](#)

[The UNIX TimeSharing System Dennis M. Ritchie and Ken Thompson - Bell Laboratories](#)

[The Design and Implementation of the FreeBSD Operating System \(2nd Edition\) \(ISBN-13: 978-0321968975\)](#)

[Wikipedia : Unix/Linux init](#)

[Wikipedia : Unix System III \(3\)](#)

[Wikipedia : UNIX System V \(5\)](#)

[Wikipedia : CSRG \(Computer Systems Research Group\)](#)

https://www.opensource-osijek.org/dokuwiki/wiki:knjige:uvod_u_linux : Uvod u Linux i Linux napredno

<https://www.freebsd.org/doc/en/articles/linux-users/index.html> : FreeBSD Quick Start

<http://haw.nsk.hr/arhiva/vol4/2240/37649/www.linuxzasve.com/kako-je-nastao-linux.html> : Kako je nastao Linux

http://wikilivres.ca/wiki/Twenty_Years_of_Berkeley_Unix:_From_AT%26T-Owned_to_Freely_Redistributable : BSD Unix - povijest

https://en.wikipedia.org/wiki/Sun_Microsystems : Sun Microsystems

[https://en.wikipedia.org/wiki/Solaris_\(operating_system\)](https://en.wikipedia.org/wiki/Solaris_(operating_system)) : Sun Solaris OS

Izvor svih ostalih informacija su :

<https://en.wikipedia.org> - Wikipedia - EN

i

<https://hr.wikipedia.org> - Wikipedia - HR

Više informacija o knjizi

Izvorne inačice knjige u **PDF** i **DokuWiki** formatu možete dobiti na stranici :

<https://www.opensource-osijek.org/wordpress/kratka-povijest-unixa-od-unicsa-do-freebsd-i-linuxa/>

QR Code - Link na stranicu:



Upoznati ćemo se s povijesti razvoja operativnog sustava Unix, kao i njegovih raznih varijanti. Vidjet ćete koliko je entuzijazam ljudi bio važan te koliko su razne, gotovo odvojene priče, na kraju isprepletene i povezane. Govorit ćemo i o pojedincima i grupama ljudi koji su omogućili nastanak *Unixa*, te raznih tehnologija, koje su bile nužne i za njegov nastanak ali i dalji razvoj. Zanimljivo je što se radi o istim tehnologijama, koje su danas u upotrebi i u svim drugim operativnim sustavima.

Dodatno, spomenuti ćemo sveučilišta i razne vizionarske tvrtke te njihov doprinos u razvoju *Unixa* i svijeta u kojem danas živimo.

ISBN : 978-953-59438-0-8

Poveznica na online verziju knjige

