

# Zgodovinska vloga operacijskega sistema VMS pri razvoju COBISS-a

## The historical role of the VMS operating system in the development of COBISS

Davor Šoštarič<sup>1</sup>

**IZVLEČEK:** V prispevku se z nekaj zgodovinske distance spomnimo na razmere, ki so pripeljale do tega, da je operacijski sistem VMS postal podlaga za potrebe razvoja in implementacije nacionalnega vzajemnega bibliografskega sistema COBISS. Najprej spoznamo nekatere bistvene značilnosti operacijskega sistema VMS, nato pa spremljamo pot njegove transformacije v tej podporni vlogi za COBISS, vključno s kasnejšo opustitvijo.

**KLJUČNE BESEDE:** operacijski sistem, VMS, OpenVMS, VAX/VMS, COBISS

**ABSTRACT:** The paper reviews, with some historical distance, the circumstances leading to the VMS operating system becoming the basis for developing and implementing the COBISS national shared bibliographic system. Following a description of the basic characteristics of VMS, the paper shows the path of its transformation in its supportive role to COBISS, as well as its omission later on.

**KEYWORDS:** operating system, VMS, OpenVMS, VAX/VMS, COBISS

---

## 1 Uvod

Skupno sistemsko programsko opremo, ki je odgovorna za delovanje celotnega računalnika (strojna, programska, aplikativna in komunikacijska oprema), obenem pa skrbi za skupne vire in servise za računalniške programe in interakcijo z okolico, označujemo z izrazom **operacijski sistem**. Praviloma je imela vsaka vrsta računalnikov svoj operacijski sistem; le zelo sorodni računalniki iz iste ali sorodne družine posameznega proizvajalca so uporabljali istega. Kasneje je ta izključna navezanost med računalnikom in operacijskim sistemom sicer izginila, saj so nekateri proizvajalci začeli izdelovati računalnike, na katere so nameščali operacijske sisteme neodvisnih proizvajalcev. Vendar pa v času, ko se je v Mariboru vizionarska ekipa entuziastov začela ukvarjati s knjižničnimi informacijskimi sistemi, iz katerih je kasneje zrasel danes vsem znani COBISS, ni bilo možno razmišljati o ločenih, samostojnih poteh računalnika in operacijskega sistema.

V začetku osemdesetih let prejšnjega stoletja se je v Sloveniji začel krhati na videz nedotakljivi imperij svetovnega računalniškega giganta, ki je praktično do zadnje pore obvladoval slovensko računalniško sceno. Vsaj dva slovenska predstavnika sta začela oglaševati »svojo« proizvodno linijo, čeprav je šlo le za licenčno kopiranje in lepljenje slovenskih imen na sicer znane tuje blagovne znamke. Zelo ugodna tržna pravila za akademska okolja so pripeljala računalnike z operacijskim sistemom VMS tudi na Univerzo v Mariboru in v njen novoustanovljeni Računalniški center Univerze v Mariboru (RCUM). Iz njega izhajajoči kasnejši samostojni javni zavod IZUM je tej informacijskotehnološki usmeritvi v začetku devetdesetih sledil.

---

<sup>1</sup> Davor Šoštarič, Institut informacijskih znanosti (IZUM), Maribor, Slovenija, [davor.sostaric@izum.si](mailto:davor.sostaric@izum.si).

RCUM in za njim IZUM sta tako na svojih računalniških sistemih praktično od začetka za vse svoje potrebe uporabljala operacijski sistem VMS. V letu 2020 IZUM ugaša še zadnje produkcijske aplikacije, ki so tekle na tem sistemu. Trenutno je operacijski sistem VMS v IZUM-u začasno v uporabi le še za nekatere neizogibne specialne aplikacije, nadzor podatkovne integritete, začasne podporne postopke, kontinuiteto sistemov COBISS zunaj Slovenije in statistične izračune; z zaključkom leta 2020 bo po vsej verjetnosti tudi tega konec. Obdobje skoraj štiridesetih let zvestobe svetovni uspešnici bo s tem zaokroženo. Vsekakor je bilo to obdobje zelo uspešno.

## 2 Značilnosti operacijskega sistema VMS

Družina 32-bitnih računalnikov VAX ameriškega proizvajalca Digital Equipment Corporation (DEC) je bila zasnovana na posebni arhitekturi z novimi neodvisnimi operatorji, podatkovnimi tipi, mehanizmi naslavljanja in nabori ukazov, ki jih drugi njihovi računalniki niso poznali. Na omenjenih računalnikih VAX je tekel operacijski sistem VMS. V skladu z različnimi poimenovanji procesorjev in novimi generacijami se je tudi ime tega operacijskega sistema pojavljalo v več različicah: VAX/VMS, OpenVMS pa tudi samo VMS. V tem prispevku bomo uporabljali izraz operacijski sistem VMS.

Pri obeh nazivih – VAX za računalnik in VMS za operacijski sistem – je ključnega pomena črka **V**, ki je kratica za besedo *virtual* (navidezni). Celotno ime VMS pomeni *Virtual Management System*, torej sistem navideznega upravljanja. Upravljanje ni v resnici čisto nič navidezno; mišljeno je *upravljanje navideznega naslovnega prostora*. Operacijski sistem VMS namreč zna za vsakega uporabnika sproti naslavljanju (in s tem uporabljati) zelo velik pomnilniški naslovni prostor; več kot štiri milijarde (od 0 do  $2^{32} - 1$ ) naslovov obsega *virtualni naslovni prostor*, računalniška arhitektura pa je omogočala hitre in zanesljive preslikave med fizičnim pomnilnikom in diskovnim prostorom za shranjevanje pomnilniških vsebin, ki začasno nimajo prostora v fizičnem pomnilniku. Vsakemu uporabniku je na voljo njegov celotni navidezni (*virtualni*) prostor. Uporabnik se ne ukvarja z delitvijo na realni in navidezni del. Zanj je vse en sam enotni pomnilniški prostor. Operacijski sistem VMS zna poskrbeti, da so pomnilniške vsebine najprej ustrezno razporejene na večje enote – strani, v postopku njihovih obdelav pa uporabniku neopazno prehajajo iz fizičnega pomnilnika v prostor na disku in obratno.

Domorodni nabor ukazov (angl. *native instruction set*) v splošnem ni karakteristika operacijskega sistema, ampak je vezan na procesor (v našem primeru prvotno VAX, kasneje Alpha in Itanium). Tu ga omenjamo le zaradi popolnejše slike. Pri procesorjih VAX je šlo za zelo obogaten nabor ukazov, kasnejša Alpha in Itanium pa že uporabljata omejen nabor ukazov (angl. *RISC – Reduced Instruction Set Computing*), ki je prišel v uporabo pri skoraj vseh kasnejših procesorjih na tržišču. V svoji osnovni različici pozna šest primarnih podatkovnih tipov operandov, skupaj z dodatnimi variacijami obstaja skupno 14 (z upoštevanjem predznačenosti pa celo 19) podatkovnih tipov različnih velikosti, natančnosti in obsegov. Osnovna računalniška beseda je velika 32 bitov na procesorjih VAX ter 64 bitov na procesorjih Alpha in Itanium. Na voljo je 16 registrov (enake velikosti kot osnovna beseda). Nekateri med njimi imajo posebno vlogo (programski števec, kazalec sklada, kazalec seznama argumentov, status procesorja ...) in jih programer lahko uporabi neposredno za konkretne namene. Ukazi so praviloma zelo intuitivno razumljivi, med njimi pa so tudi dokaj nenavadni ukazi v primerjavi z ukazi v drugih sistemih (npr. ukaz za neposredno izračunavanje vrednosti polinoma, selektivni stavki, ukazi za delo z velikimi kosi pomnilnika naenkrat ...). Osnovne operacije nad

posameznimi operandi so praviloma odvisne od podatkovnega tipa, ukazi za vejanje, pogojne in brezpogojne skoke ter za operacije nad biti in bitnimi polji pa so na pogled zelo podobni tistim, ki jih srečamo v drugih sistemih.

Tukaj srečamo praktično vse svetovno razširjene programske jezike tistega časa: fortran, cobol, PL/I, RPG, pascal, algol, basic, C, lisp ... Operacijski sistem VMS je imel tudi nekatere bolj ali manj lastne programske jezike, ki so bili zunaj njegovega okolja oziroma zunaj okolja sorodnih računalnikov istega proizvajalca zelo redki, npr. zbirnik in bliss. Vsem je bilo skupno to, da so se lahko med seboj poljubno klicali in uporabljali enotne mehanizme prenosa parametrov oziroma argumentov. Ne glede na jezik izvorne kode so prevajalniki programerjeve izdelke predelali v *objektni modul*, pri katerem se je praviloma izgubila sled izvornega programskega jezika. Načeloma bi torej algoritem, sprogramiran v dveh različnih programskih jezikih, dal skoraj identična objektna modula. Več objektnih modulov med seboj in z vnaprej pripravljenimi moduli sistemskih knjižnic je poseben program (angl. *linker*) povezal v izvršljivo kodo (angl. *executable image*).

Operacijski sistem VMS je tudi večopravilen, paketno in transakcijsko orientiran ter omogoča delitev istega računalniškega vira med več uporabnikov (angl. *time-sharing*).

### 3 Prednosti za COBISS

Med prvimi informacijskimi podporami, ki jih je RCUM začel razvijati za potrebe mariborske univerze, je bila tudi izgradnja knjižničnega informacijskega sistema. Glede na vse že akumulirano znanje, jasno načrtano podporno pot jedrnega razvoja informacijske tehnologije na obeh takratnih slovenskih univerzah in ekonomske diktate računalniške nove dobe je bila odločitev o podlagi operacijskega sistema VMS za predhodnika nacionalnega vzajemnega bibliografskega sistema COBISS popolnoma naravna. Ni šlo torej za nikakršno z analizami podkrepljeno zavestno odločitev, ampak se je tako zgodilo le zaradi takratnega stanja.

Preden si ogledamo pogloblitve prednosti operacijskega sistema VMS, ki smo jih prepoznali kot posebej pomembne pri razvoju sistema COBISS, omenimo še dve navidezno oteževalni okoliščini za odločitev za uporabo prav tega operacijskega sistema. Operacijski sistem VMS ni imel lastnega sistema za upravljanje z bazami podatkov. Iz zgodovinskih razlogov je bil v slovenskem okolju in tudi pri nas razširjen mrežni sistem baz podatkov TOTAL ameriškega proizvajalca Cincom Systems (v kasnejših različicah ULTRA in SUPRA – sistem SUPRA se je že spogledoval z relacijskim modelom), tako da ni čudno, da so tudi začetki sistema COBISS delovali na tej platformi. Slovenski distributer računalnikov, na katerih je tekel operacijski sistem VMS, je razvil lastni produkt za upravljanje baz podatkov, vendar so testiranja pokazala, da za naše namene ne zadošča. Kmalu se je pokazalo, da nobeden od nam dostopnih sistemov za upravljanje baz podatkov ne ustreza. Vsi sistemi so namreč izhajali iz klasične poslovne informatike, mi pa smo se srečevali z zakonitostmi, ki so bile s temi orodji zelo težko obvladljive. Tu mislimo predvsem na številskemu svetu podrejene podatkovne strukture, pri svojem delu pa smo imeli opravka z nizi znakov zelo spremenljivih dolžin. Seveda so obstajale svetovne ustrezne komercialne rešitve, vendar so nam bile cenovno nedostopne. Zaradi tega in tudi zaradi odličnega internega sistema za delo z datotečnimi zapisi smo se odločili, da razvoja sistema COBISS ne vežemo na noben uveljavljeni sistem za upravljanje baz podatkov, ampak da naredimo svojega, ki bo temeljil na standardiziranem datotečnem sistemu. Temu načelu smo ostali zvesti precej dolgo.

Podobna zgodba se je odvijala na področju programskih modulov za vnos podatkov (angl. *screen management services*). Uveljavljena, a za naše namene že zastarela orodja za vnos in manipulacijo podatkov oziroma interakcijo med človekom in programskim okoljem smo nadomestili s povsem novimi sistemi, ki so bili plod lastnega znanja in razvoja.

Katere so torej pglavitne prednosti operacijskega sistema VMS, ki so se nam zdele posebej pomembne pri razvoju sistema COBISS? To so:

### **3.1 Učinkovita razpršenost delovnih mest (cenovno zelo ugodno terminalsko omrežje)**

Poslovna informatika se je v zgodnjih osemdesetih letih prejšnjega stoletja ravno začela oddaljevati od masovnega zajemanja podatkov iz obrazcev in se približevati k računalniku prijaznejšemu mediju (kartice in trakovi) v specializiranih oddelkih za zajemanje podatkov (pog. *luknjačice*), tako da so za to odgovorni sodelavci neposredno vnašali podatke v računalniški sistem prek namenskih enot (terminali). Hkrati se je zgodil premik iz tipičnih *paketnih obdelav* (naknadno in praviloma zunaj običajnega delovnega časa) v sprotno (*interaktivno*) obdelavo podatkov. Zato je bilo potrebno dvoje: drugačna zasnova programskih algoritmov in programov ter ustrezno opremljeno delovno mesto (terminali, ponekod tudi majhni tiskalniki) neposredno pri strokovnem sodelavcu. Pri računalnikih, na katerih je tekel operacijski sistem VMS, so bili terminali razmeroma preprosti (ekran s tipkovnico brez lastnih procesorskih zmogljivosti), njihova povezava do centralnih računalniških enot pa tehnično in cenovno dokaj nezahtevna (tudi na nekoliko večje razdalje), še posebej po uveljavitvi posrednih poti (npr. pasivno omrežje s protokolom *ethernet*). Nekateri konkurenčni sistemi so temu sicer sledili, nekateri pa še dolgo ne. K ekonomskemu učinku je pripomogel tudi znaten prihranek pri človeških virih.

### **3.2 Sobivanje različnih programskih jezikov z enotnim principom delovanja**

Večina takratnih računalniških sistemov pri nas je preferirala enega ali kvečjemu dva bolj razširjena splošna programska jezika (npr. fortran ali pascal za naravoslovne in tehniške probleme ter cobol ali PL/I za poslovne probleme). Vsak računalniški sistem je imel tudi ekskluzivni sebi lasten zbirni jezik (angl. *assembler*), ki je zahteval zelo specifično izobraževanje in poznavanje jedrnih internih struktur operacijskega sistema. Na računalnikih z operacijskim sistemom VMS so drug ob drugem brez posebnosti tekli skoraj vsi svetovno znani programski jeziki, ki so bili med seboj popolnoma kompatibilni, zamenljivi, medsebojno klicljivi in so delovali na identičnih zasnovah. Tako so bile prve verzije programske opreme za posamezne segmente sistema COBISS napisane v cobolu z dodanimi posebnimi moduli v fortranu in zbirniku, kasneje pa se je zbirnik obdržal samo še za najbolj specialne systemske module in orodja, glavno vlogo programja pa je skoraj v celoti prevzel basic.

### **3.3 Bogata dokumentacija (tudi iz virov, neodvisnih od proizvajalca)**

Poznavanje višjih programskih jezikov (vsi razen zbirnikov) je praviloma zadoščalo za klasično programiranje (kodiranje algoritmičnih zasnov). Za optimiziranje izvršljive programske kode pa je bilo koristno poznati notranji ustroj računalniške arhitekture in operacijskega sistema VMS. Govorimo o *internih gradnikih*; le-ti se tukaj pojavljajo v natančno oblikovanih *podatkovnih strukturah*, ki so specifična operacijskega sistema VMS. Za razliko od večine takratnih računalniških sistemov v naši okolici to znanje ni bila izključna domena posvečenih. Že proizvajalec je dal na voljo ogromno odličnih priročnikov, osnovna uradna priročniška dokumentacija pa je v svojih dveh najobsežnejših izdajah (»siva« in »oranžna«) pred

prehodom na izključno e-obliko presejala dolžino štirih metrov (tekoča postavitve knjige ob knjigo). Tudi slovenski distributer operacijskega sistema VMS je prirejal odlične tečaje in seminarje z izvrstnim in obsežnim spremljajočim gradivom.

Zaradi proizvajalčeve politike odprtosti in neekskluzivnosti so številni založniki (predvsem v ZDA) izdajali kopico zelo pomembnih priročnikov in vrhunskih knjig s področja razkrivanja internih skrivnosti operacijskega sistema VMS. V našem okolju (RCUM in IZUM) smo poleg uradne literature pridno posegali po predvsem naslednjih knjigah (slika 1): *VAX Architecture Handbook* (Digital Equipment Corporation, 1981), *VAX-11 Assembly Language Programming* (Baase, 1983) in *VAX/VMS Internals and Data Structures* (Goldenberg in Kenah, 1991). Prva knjiga prihaja sicer iz založniške produkcije proizvajalca računalnikov, na katerih je tekel operacijski sistem VMS, drugi dve pa iz neodvisnih krogov. Vse tri so imele v RCUM-u in IZUM-u skoraj ikonični status in še danes zavzemajo posebno mesto v avtorjevi osebni in IZUM-ovi knjižnici.



Slika 1: Najpomembnejši viri znanja za VMS (Vir: osebni arhiv, 1982–2020)

### 3.4 Zаметki omrežnih aplikacij

Zamislil o povezovanju računalnikov so le malo mlajše od računalnikov samih, vendar se je fokus uporabnosti teh idej močno spremenil. V začetku je šlo za težnjo po večji procesorski moči in izmenjavi podatkov ter boljšo izkoriščenost podatkovnih kapacitet, verjetno pa tudi še za rezervni načrt za okrevanje po hujšem izpadu. Vsekakor je bil pogled na omrežje povsem drugačen kot danes (tudi za izraz »omrežje« se je uporabljala danes nezaželena »mreža«).

Zasnova operacijskega sistema VMS je že od začetka presegala te poglede. Omogočala je neposredno interakcijo med programi v realnem izvajalnem času. Enotni sistem naslavljanja podatkovnih virov je poleg diskovne enote, mape in imena datoteke v ožjem smislu predvideval tudi ime omrežnega gostitelja. Nekaj podobnega so sicer poznali konkurenčni računalniški in operacijski sistemi, in tudi VMS ni bil izjema v tem, da je omrežje moralo biti homogeno. Možnost povezovanja virov na različnih sistemih je bila še vedno iluzija. Lepota lastnega koncepta računalniškega omrežja (prvotna blagovna znamka je nosila ime *DECnet*) je bila tudi v tem, da je močno ustrezala teoretičnemu sedemnivojskemu referenčnemu modelu (*OSI*), ki je kasneje postal standard in so mu sledile praktično vse komercialno uspešne različice omrežnih konceptov (ne nazadnje tudi različice omrežij po protokolih TCP/IP, vključno z internetom). Ne glede na naravno delitev sedemnivojskega referenčnega modela na tri skupine plasti (medijska, gostiteljska in aplikativna raven) je bila moč VMS-ovske implementacije prav v najvišji (aplikativni) ravni. Čeprav terminološko nesistematičen in zmeden je operacijski sistem VMS omogočal elegantno komunikacijo med programi najprej znotraj domorodnega omrežja DECnet, kasneje pa tudi po sodelujočih omrežjih, ki so upoštevale skupne de facto standarde.

### 3.5 Prefinjen datotečni sistem in delo z datotečnimi zapisi

Operacijski sistem VMS je uporabljal datotečni sistem z nazivom *Files-11*. Že iz dela imena datotek (tip) se je dalo sklepati na vsebino in spekter predvidenih tipov datotek je bil zelo širok. Struktura osnovnih komponent datotečnega opisa (angl. *file header*) je bila enotna ne glede na predvideno uporabo, medij ali karakteristike strojne opreme.

Pravo bogastvo pa je predstavljal sistem za delo z datotečnimi zapisi (*RMS – Record Management System*), ki je s svojimi bogatimi knjižnicami klicljivih modulov omogočal elegantno delo neposredno tudi iz višjih programskih jezikov. Široka paleta kombinacij datotečne organizacije s sistemom datotečnih zapisov in sistemom zaklepanja zapisov (angl. *lock management*) je načrtovalcem in programerjem dajala skoraj neomejene možnosti.

## 4 Razlogi za opustitev

Po približno četrto stoletja sožitja je operacijski sistem VMS kljub novim verzijam začel ovirati nadaljnji razvoj sistema COBISS. Večino tega, kar je v prejšnjem poglavju omenjeno kot izrazita prednost, so začeli ponujati tudi drugi operacijski sistemi. Proizvajalci računalnikov, na katerih je deloval operacijski sistem VMS, so se združevali v korporacije in razdruževali, njihov tržni delež v svetovnem merilu je doživljal pretrese, konkurenca ni počivala, pa tudi cenovna politika se je spremenila, tako da nismo bili več deležni nekdanjih velikih ugodnosti. Največji udarec pa je predstavljal upad zagona pri razvoju operacijskega sistema VMS. Sledenje razvojnim trendom, novostim in strokovnim dognanjem je začelo močno zaostajati. Nove funkcionalnosti, ki so bile pri drugih operacijskih sistemih naravne, so pri operacijskem sistemu VMS delovale le z nesistematičnim dopolnjevanjem systemske programske opreme. Zelo težko je bilo slediti eksploziji novih komponent strojne opreme, ustrezno podpirati vedno nove večpredstavne vmesnike in uporabljati nove programerske pristope, vključno s programskimi jeziki. Tudi drugje po Sloveniji so se začele migracije informacijskih sistemov na druge platforme, predvsem na različice Unix in na Microsoftov Windows. Posledično je bilo tudi ustreznih strokovnjakov vse manj v naši bližini, programerske veščine za obvladovanje okolij,

povezanih z operacijskim sistemom VMS, pa se v slovenskih izobraževalnih procesih niso več negovale.

Prodor osebnih računalnikov je na smetišče tehnološke zgodovine odplaval alfanumerične terminale, ki niso več zadoščali niti za najosnovnejše posodobitve informacijskega okolja na delovnih mestih v knjižnicah. IZUM se je soočal z vedno pogostejšimi zahtevami po spremembah nekaterih osnovnih konceptov pri avtomatizaciji knjižničnih procesov in strokovnih izzivih. Pasivni odjemalci (terminali) niso omogočali aktivnejše vloge pri lokalnih obdelavah in načelo »vse je sprogramirano vnaprej« je tudi ob obilni parametrizaciji onemogočalo propulzivnejši razvoj. Nova generacija programske opreme COBISS (popularno imenovana COBISS3) je predvidevala take funkcionalnosti, ki jih je bilo v tistem trenutku z že zastarelim pristopom možno implementirati le z obilo napora.

Naprej ni šlo več. Posamezne komponente sistema COBISS so bile predelane na način, ki ni determiniral informacijskega okolja ne s sistemsko programsko opremo ne s strojno opremo. Na srečo je bila programska oprema sistema COBISS že od samega začetka zasnovana zelo modularno, tako da je IZUM zadnjih nekaj let lahko postopoma v posameznih segmentih opuščal operacijski sistem VMS in ga polagoma nadomeščal z drugimi koncepti. Lasten sistem za upravljanje baz podatkov je bil v določenih segmentih nadomeščen s komercialnim, čeprav smo se zavedali, da gre le za začasno premostitev najbolj perečih težav. Pri programiranju smo se vse bolj naslanjali na nove programerske principe in jezike, prilagodili smo interne standarde in vpeljali nove metodologije pri organizaciji, načrtovanju in testiranju. Rezultati niso izostali, vendar je operacijski sistem VMS še vedno kraljeval v ozadju.

IZUM se je zavedal, da bo prehod na novo informacijsko okolje nujen. Hkrati je bilo jasno, da se je nanj treba zelo dobro pripraviti. Najbolj kritično vprašanje je bilo, kakšen naj bo novi sistem, na katerem bo COBISS tekel vsaj enako dobro, obenem pa bodo ostala vrata dovolj odprta za več kot le kratkoročni nadaljnji razvoj. Ob nekajletnem obdobju analiziranja in načrtovanja, včasih stopicanja na mestu ali celo vrtenja v krogu, pa tudi vrnitve s poti, ki so se izkazale za slepe, in ponovnega poskusa preboja, je IZUM še vedno redno nadgrajeval obstoječi sistem COBISS3 in ga dopolnjeval. Slika, kaj hočemo, je bila dokaj jasna, pot do uspešne izvedbe pa zapletena.

## 5 Proces prehoda v sedanje okolje

Prehod iz operacijskega sistema VMS ni bil premočrten. Najprej so se novosti pojavile na strežniški strani, in sicer z uporabo strežnikov z operacijskima sistemoma Windows in Linux, nato so sledili aplikacijski strežniki platforme Java EE.

Za shranjevanje podatkov se je v zaledju, po neposrečeni<sup>2</sup> fazi implementacije posebnega modela organizacije podatkov, ki je nekje vmes med datotečnim sistemom in relacijskimi sistemi za upravljanje baz podatkov začela na strežnikih z operacijskim sistemom Linux uporabljati relacijsko bazo podatkov Oracle. Tudi na strani odjemalca so se z uporabo aplikacije na osnovi programskega jezika java lahko za delovna mesta v knjižnicah uporabljali osebni ali prenosni računalniki z različnimi operacijskimi sistemi (večinoma Windows), nekaj pa tudi Linux in MacOS.

---

<sup>2</sup> Osebno avtorjevo mnenje.

Prva segmenta programske opreme »po novem« sta bila COBISS3/Medknjižnična izposoja in COBISS3/Upravljanje aplikacij. Pri tem se je hitro pokazala nujnost gradnje ogrodja za razvoj aplikacij, ki bi omogočalo bistveno učinkovitejšo razvojno pot številnih razmeroma kompleksnih modulov med seboj zelo povezanih lokalnih aplikacij COBISS. Ogrodje je bilo razvito že v novem programskem okolju (java) in se je izkazalo za ključno prebojno orodje pri nadaljnjih programskih modulih in segmentih.

Pri načrtovanju prehoda programske opreme na drugačne osnove so prednost imeli segmenti, ki v obstoječem okolju še niso bili podprti. Najprej so bili na vrsti COBISS3/Nabava, COBISS3/Serijske publikacije in COBISS3/Zaloga s shranjevanjem zaloge, vzporedno z njimi pa COBISS3/Izpisi, ki so knjižničarjem omogočili predvsem parametrizirano pripravo obrazcev, njihovo povezovanje s podatki v bazah podatkov in pripravo izpisov v različnih izhodnih formatih.

Sledila sta najzahtevnejša segmenta: COBISS3/Katalogizacija in COBISS3/Izposoja. Razvoj nove generacije katalogizacije je bil zastavljen tako, da ni bilo postopnega prehoda, zato ga je bilo možno razmeroma hitro vključiti v produkcijo oziroma preklopiti iz starega načina dela v novi. Po uspešni implementaciji v IZUM-ovi knjižnici je leta 2012 nova katalogizacija uspešno stekla v Narodni in univerzitetni knjižnici. Posamezne knjižnice so jo osvajale postopoma po natančnem načrtu in ob polovici leta 2016 so vse knjižnice v Sloveniji svoje gradivo katalogizirale izključno v segmentu COBISS3/Katalogizacija.

Pri izposoji smo ubrali drugačno pot. Najprej smo podprli knjižnice, ki potrebujejo najmanj funkcionalnosti v postopku avtomatizirane izposoje, v kasnejših fazah pa smo vključevali vedno zahtevnejše funkcionalnosti za različne tipe knjižnic in za različno raven njihovih zahtev. Tako so šele na koncu prišle na vrsto splošnoizobraževalne knjižnice, ki so imele s stališča izposoje največje zahteve in so bile tudi najzahtevnejše za operativno izvedbo prehoda iz starega v novo okolje.

Potrebovali smo nove komponente za prikaz bibliografskih podatkov. Njihov razvoj je omogočil tudi realizacijo novih verzij bibliografij.

Vpeljevanje sofisticiranega sodobnega iskalnika je bil posebno trd oreh. Novi COBISS+, ki je razvit na platformi Java, ni zgolj nadomestil nekdanje uspešne in široko uporabljane spletne aplikacije COBISS/OPAC. Z mislijo na kasnejšo integracijo s ponudniki integralnih indeksov elektronskih virov smo za že obstoječe vsebine iskali načine za obogatitev zmožnosti z novimi paradigmami, kot so fasetno iskanje, urejanje po relevantnosti, iskanje po več virih hkrati, krnilnik, upoštevanje funkcionalnih zahtev za bibliografske zapise itd. Po intenzivnih testiranjih in lokalnih prototipnih različicah smo se odločili za nabavo odprtokodnega iskalnika Lucene/SOLR, ki je izpolnil vsa naša pričakovanja.

Pri spoštovanju vredni dolžini življenjskega cikla programske opreme COBISS, ki je tekla na operacijskem sistemu VMS, je nastala cela vrsta pomožnih modulov in orodij (različni izpisi, poročila, statistike ...), za katere je bilo treba najti ustrezno nadomestilo. Šele ko je bilo to urejeno, je bilo možno »pritisniti na gumb« in ugasniti operacijski sistem VMS.

Ostala sta še vprašanje sistema za upravljanje baz podatkov in generalna zamenjava operacijskega sistema. Pri bazah podatkov smo se odločili za zamenjavo relacijske baze Oracle



z odprtokodnim sistemom PostgreSQL. Produkcija za končne uporabnike se seli na operacijski sistem iz družine Unix; konkretno gre za CentOS (ena od implementacij Linuxa).

## 6 Sklepna misel

Operacijski sistem VMS je bil mnoga leta ena od temeljnih podlag računalniškega razvoja v Sloveniji. Generacijam študentov je bil eden od najpopolnejših vpogledov v principe delovanja računalnikov (izven ravni elektronike). Deloval je na računalniških sistemih univerz, raziskovalnih inštitutov in raziskovalnih organizacij, pa tudi v gospodarstvu ni bil redek. V predinternetni dobi je bil glavni povezovalni faktor neformalnega omreženja slovenskih računalnikov.

*Tabela 1: Razvojna pot osnovnih elementov podlag, na katerih stoji in deluje COBISS*

Komponenta	Na začetku	Potem	Danes
operacijski sistem	VMS	VMS; MS Windows	CentOS
baza podatkov	Total/Ultra/Supra	lasten sistem; Oracle	PostgreSQL
programski jezik	cobol; basic; zbirnik		java

Več kot 35 let je bil operacijski sistem VMS podlaga za razvoj in delovanje nacionalnega vzajemnega bibliografskega sistema COBISS. V zadnjih letih pa ni več zagotavljal zadostne podpore za nove generacije vsebinskih in računalniških zahtev in je postal resna ovira na nadaljnji razvojni poti. Te ovire sicer niso nepremostljive, zahtevajo pa ogromna finančna vlaganja, ki si jih ne IZUM ne knjižnična skupnost ne moreta privoščiti. Rešitev za spremembo se je iskala dolgo in odgovorno konservativno, pot do sprememb ni bila premočrna. Tabela 1 povzema v prispevku opisano razvojno pot komponent z začetnim, vmesnim in končnim stanjem.

## Zahvala

Na tem mestu bi se rad zahvalil nekdanjim in sedanjim sodelavcem in pionirjem dela v operacijskem sistemu VMS, ki so prebrali posamezne dele tega prispevka in me s svojimi pripombami in predlogi spodbujali, pa tudi obvarovali kakšne nedoslednosti. Če se nam je kljub vsemu kakšna izmuznila, gre na moj račun. Navajam po abecednem redu: Saša Bibič, Slavko Breznik, Domen Šetar, Bojan Štok, Robert Vehovec.

## Reference

Digital Equipment Corporation, 1981. *VAX Architecture Handbook*. Maynard, ZDA: Digital Equipment Corporation.

Baase, S., 1983. *VAX-11 Assembly Language Programming*. Englewood Cliffs, ZDA: Prentice-Hall Inc.

Goldenberg, R. in Kenah, L., 1991. *VMA/VMS Internals and Data Structures: version 5.2*, Woods Drive, ZDA: Digital Press.

Osebni arhiv avtorja, 1982–2020.