Izrada modela sklopovlja računalne platforme upotrebom alata HWLOC

Štadler, Hrvoje

Undergraduate thesis / Završni rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:400796

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-23**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Stručni studij

Izrada modela sklopovlja računalne platforme upotrebom alata HWLOC

Završni rad

Hrvoje Štadler

Osijek, 2016.

SADRŽAJ

1. UVOD 1
1.1. Zadatak završnog rada 2
2. Računalni sustav i sklopovlje 3
2.1. Operacijski sustav 3
2.2. Sklopovlje i programska podrška 4
2.2.1. Sklopovlje 4
2.2.2. Programska podrška 5
2.3. Procesor 5
2.4. NUMA i priručna memorija 5
3. HWLOC (HARDWARE LOCALITY) 8
3.1. Paralelno programiranje 8
3.2. Paralelne aplikacije na podijeljenom sklopovlju 9
3.2.1. Model komunikacije dijeljenom memorijom 9
3.2.2. Model komunikacije razmjenom poruka 10
3.2.3. Model komunikacije podatkovnim paralelizmom 10
3.3. Instalacija i upotreba HWLOC-a – Microsoft Windows 10
3.4. Instalacija i upotreba HWLOC-a – Linux 11
4. CPU-Z 17
4.1. Instalacija i upotreba – Microsoft Windows 17
4.2. Instalacija i upotreba – Android 18
5. AIDA64 20
5.1. Instalacija i upotreba – Microsoft Windows 20
5.2. Instalacija i upotreba – Android 21
6. USPOREDBA ALATA 23
7. ZAKLJUČAK 24
LITEATURA 25
SAŽETAK 26
ABSTRACT 27
ŽIVOTOPIS 28

1. UVOD

Tema završnog rada je „Izrada modela sklopovlja računalne platforme upotrebom alata HWLOC“. Svatko tko posjeduje osobno računalo, mobitel ili tablet zna da se u tom njihovom uređaju nalazi složena struktura različitih čipova i sklopova koja im spojena u cjelinu i upravljana raznim programima omogućava jednostavno korištenje. No, ukoliko se želi detaljnije proučavati struktura sklopovlja računala, u obzir svakako treba uzeti instaliranje nekih od alata za inspekciju sklopovlja. Primjerice, programeri moraju u obzir uzeti sklopovlje i njihove karakteristike prije nego pokušaju iskoristiti prave performanse određenog dijela sklopovlja. S vremenom postaje sve teže pronaći način kako korisniku prikazati arhitekturu sklopovlja pojedinog uređaja uzevši u obzir brzinu kojom se one razvijaju. Tu se javlja potreba za alatima za inspekciju sklopovlja koji mogu pratiti taj razvoj, a jedan od njih sasvim sigurno je HWLOC. Takvi alati korisnicima pružaju jednostavan, a opet vrlo detaljan prikaz arhitekture sklopovlja i pobliže im predstavljaju uređaj kao cjelinu.

1.1. Zadatak završnog rada

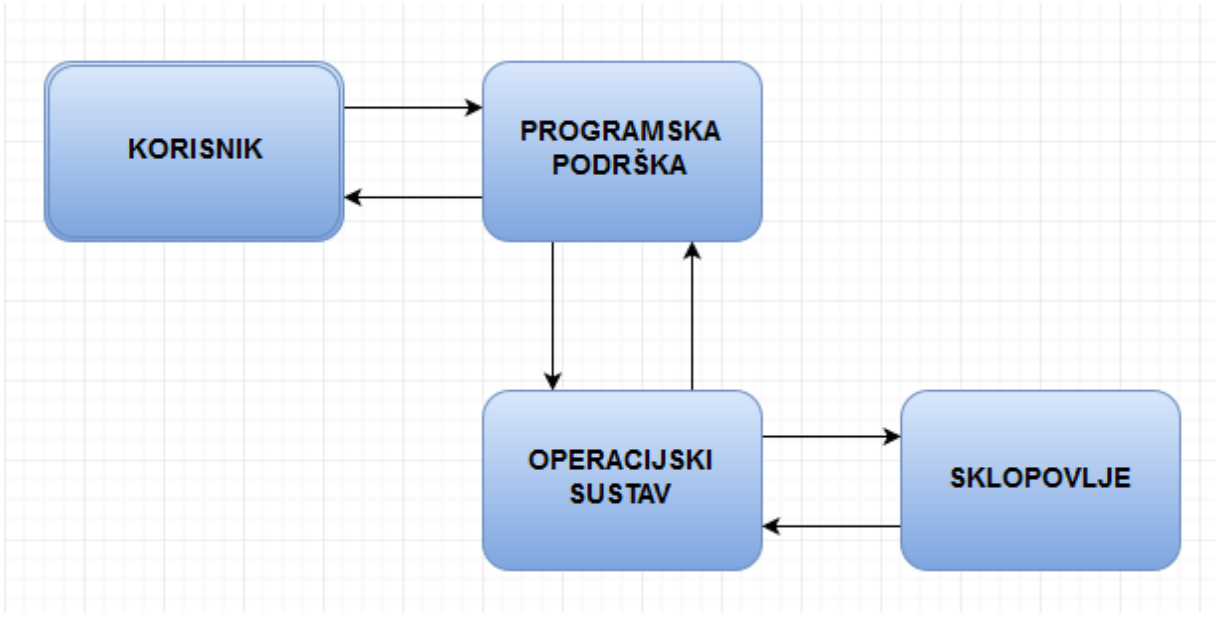
HWLOC (eng. *Hardware locality* – lokalitet sklopovlja) je glavni predstavnik skupine alata koji prikazuju odradbenu i memorijsku strukturu nekog računalnog sustava. Sa spomenutim alatom pokušat će se izraditi virtualna platforma uz opis pojedinih naredbi te pokušati prikazati i opisati postupak koji se vrši prilikom izrade istih. Uz HWLOC, istražit će se i njemu slični alati koji će biti testirani na različitim računalnim sklopovima i različitim operacijskim sustavima. Iznijet će se njihove mogućnosti i provesti razni testni slučajevi. Opisat će se i proces instalacije, dohvat informacija na različitim platformama, usporedit će se rezultati za isprobane alate i sl.

2. Računalni sustav i sklopovlje

Alati za inspekciju računalnog sklopovlja osmišljeni su kako bi korisniku pružili uvid u sklopovlje i programsku podršku računala. Kako bi korisnik mogao s razumijevanjem koristiti tu vrstu alata, potrebno je da bude upoznat s nekim od važnijih pojmova kada je riječ o samim računalima kao što su: operacijski sustav, sklopovlje, programska podrška, procesor, NUMA, priručna memorija i sl.

2.1. Operacijski sustav

Programska podrška koja povezuje sve dijelove računalnog sustava i korisniku pruža jednostavno sučelje za upravljanje i rad s istima naziva se operacijski sustav. Operacijski sustav objedinjuje sve fizičke komponente računala kao što su ulazno-izlazni uređaji, kartice (mrežne, grafičke, zvučne...), procesori, memorije i slično. Radi na principu da se pokreće automatski. Kada korisnik uključi računalo, operacijski sustav „poziva“ pojedine uređaje i provjerava njihovu ispravnost i prisutnost. Na taj način on pruža programima mogućnost da koriste sklopovlje računala. Ukratko, uključivanjem računala korisnik zapravo pokreće operacijski sustav, a operacijski sustav nakon toga preuzima kontrolu nad cijelim računalom.



Slika 2.1: Prikaz odnosa korisnika, programske podrške, operacijskog sustava i sklopovlja

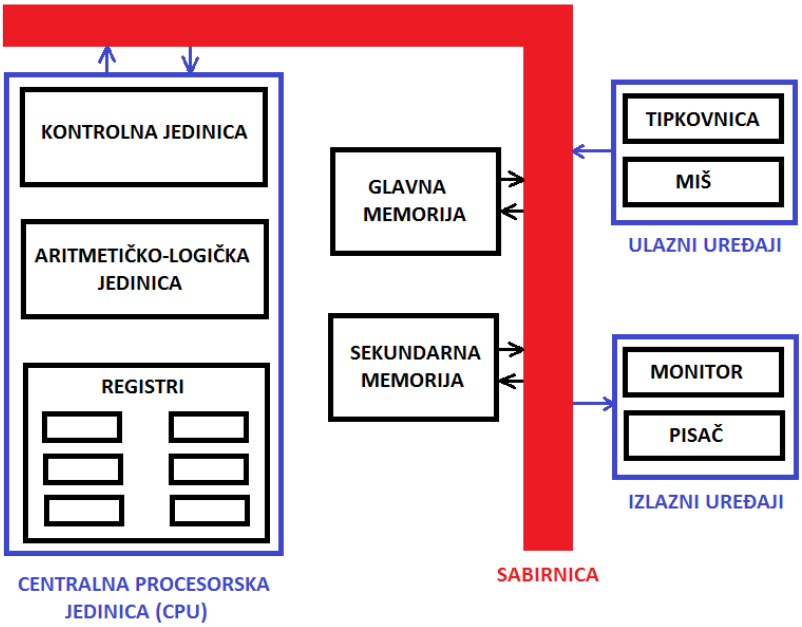
Mnoge zadaće koje operacijski sustav obavlja uključuju između ostalog i upravljanje: procesima, radnom memorijom, trajnom memorijom, ulaznim i izlaznim uređajima, imenicima, datotekama, zaštitom dijelova sustava, otkrivanjem pogrešaka u radu sustava, mrežnom povezanošću, izvješćima o greškama operacijskog sustava i sl.

2.2. Sklopovlje i programska podrška

Sklopovlje je svaka opipljiva komponenta koja se može naći u računalu, dok se programska podrška može definirati kao skup instrukcija koje mogu biti pokrenute na računalu. Te instrukcije na neki način govore računalu (sklopovlju) što činiti. Suprotno sklopovlju, programska podrška nije fizički dio računala. Računalo treba i sklopovlje i programsku podršku kako bi moglo pravilno funkcionirati. Uspoređujući ova dva pojma sa svakodnevnim životom, jedan od najboljih primjera bio bi čovjek. Kod čovjeka sklopovlje bi bilo sve što čini tijelo: kosti, mišići, koža i slično. Programska podrška bi bile misli i ideje; sve ono što govori tijelu što činiti.

2.2.1. Sklopovlje

Sklopovlje ili struktura opipljivih dijelova računala koja je temelj većini današnjih računala je Von Neumannova struktura. John Von Neumann još je u prvoj polovici 20. stoljeća osmislio arhitekturu računalnog sklopovlja kakva se uvelike primjenjuje još i danas.



Slika 2.2: Von Neumannov model sklopovlja računala

Navedena struktura sastoji se ulaznih uređaja, izlaznih uređaja, memorije i centralne procesorske jedinice, a sve je međusobno povezano sabirnicom. Pod ulazne uređaje podrazumijevaju se uređaji koji služe korisniku za unos podataka (miš, tipkovnica, mikrofon i sl.). U izlazne uređaje ubrajaju se uređaji koji korisniku pružaju neku informaciju, bilo to putem slike na monitoru ili zvuka na zvučnicima. Memorija služi za pohranu podataka (RAM pohranjuje jednokratno, a ROM pohranjuje dugoročno), a centralna procesorska jedinica zajedno sa svojim podsustavima ključna je za obradu podataka i instrukcija.

2.2.2. Programska podrška

Kao što je rečeno, centralna procesorska jedinica bavi se obradom instrukcija. Te instrukcije zapravo potječu od nekih od programa koji su instalirani na računalu. Pomoću tih programa korisnik komunicira s računalom te iznosi naredbe koje želi da se izvrše i nijedno računalo ne može funkcionirati upravo bez te programske podrške. Kako se programska podrška razvija, tako su potrebe za mogućnostima sklopovlja sve veće što ponekad zna dovesti do situacije da sklopovlje ne podržava određene programe. Kako ne bi dolazilo do takvih situacija korisnici, pa i stručnjaci koji razvijaju programe, moraju biti upućeni u mogućnosti svog sklopovlja.

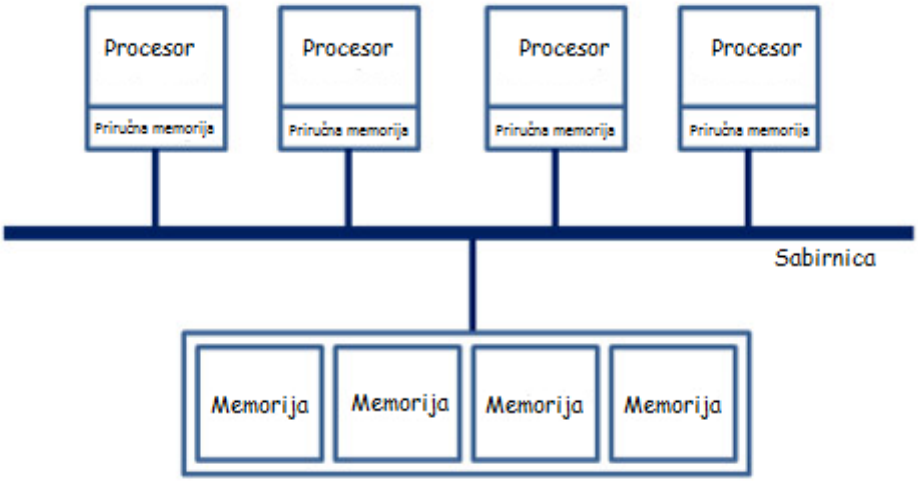
2.3. Procesor

Procesor ili CPU (eng. *Central processing unit*) središnja je i glavna komponenta računalnog sklopovlja jer upravlja svim ostalim sklopovima računala. CPU obavlja osnovne aritmetičke, logičke i kontrolerske instrukcije, a više o njima na [1]. Većina modernih CPU-a su mikroprocesori što bi značilo da se nalaze na jednom čipu s integriranim krugom. Ukoliko je procesor višejezgreni to znači da se na jednom čipu nalazi dva ili više CPU-a tj. jezgre. Brzina rada procesora, između ostalog, ovisi i o brzini radnog takta koja se mjeri u hercima i broju instrukcija u sekundi. Procesori s više jezgri u pravilu su korisniji i brži od procesora s jednom jezgrom. Iako bi bilo logično da je procesor s dvije jezgre duplo učinkovitiji od procesora s jednom jezgrom u praksi nije tako, ponajviše zbog nesavršenosti algoritama i implementacije. Što je više jezgri u procesoru to je veći obujam poslova koje procesor može u jednom trenutku obavljati.

2.4. NUMA i priručna memorija

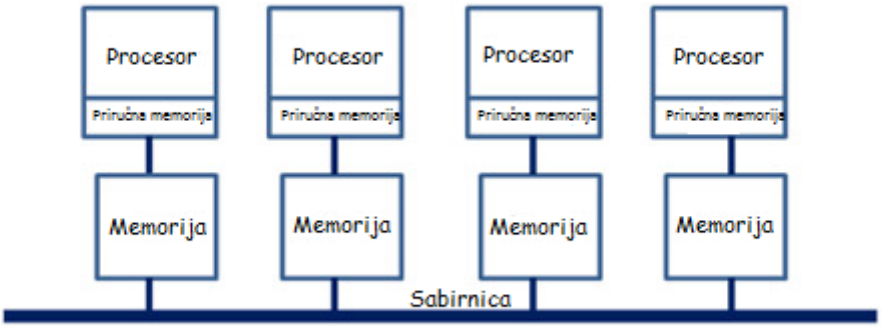
NUMA (eng. *Non-uniform* memory *access*) je memorijska arhitektura koja se upotrebljava u multiprocesiranju, gdje pristup memoriji ovisi o lokaciji memorije u odnosu na procesor. U NUMA memorijskoj arhitekturi svaki procesor ima svoju lokalnu memoriju kojoj može

pristupiti znatno velikom brzinom što, naravno, poboljšava performanse. Najbolji način za razumijevanje načina rada NUMA-e je preko njenog prethodnika: UMA (eng. *Uniform memory access*). Kod UMA memorijskog pristupa svi procesori dijele memoriju preko sabirnice (slika 2.2)



Slika 2.3: Korištenje memorije u UMA arhitekturi

Za razliku od UMA memorijske arhitekture, svaki procesor svojom sabirnicom može pristupiti svojoj memoriji (slika 2.3).



Slika 2.4: Korištenje memorije u NUMA arhitekturi

Priručna memorija je vrlo mala memorija, a u njoj se pohranjuju podaci koji se često koriste kako bi im se u budućnosti moglo lakše i brže pristupiti. Kada bi procesor pristupao RAM memoriji kako bi dohvatio neke podatke gubio bi dosta vremena na čekanje. Taj problem

riješen je ugradnjom priručne memorije na sami procesor. Na taj način procesor može puno većom brzinom pristupiti željenim podacima. Priručna memorija koja je ugrađena na procesor naziva se L1 (eng. *Level 1*) priručna memorija. Memorija koja se nalazi između L1 priručne memorije i RAM memorije naziva se L2 (eng. *Level 2*) i L3 (eng. *Level 3*) priručna memorija. Ona služi uglavnom za pohranu podataka iz RAM memorije koju bi procesor mogao uskoro koristiti.

3. HWLOC (HARDWARE LOCALITY)

HWLOC je alat koji je dizajniran od strane „Open MPI“ razvojnog tima. Članovi tog tima su doprinositelji koji imaju pravo glasanja što znači da mogu pomoći u određivanju načina na koji će se projekti izvršavati. HWLOC je jedan od njihovih najpoznatijih alata, a služi za dobavljanje hijerarhijske mape nekih od ključnih elemenata računala kao što su: memorijski čvorovi, priručna memorija, procesorske jezgre, logički procesori ili niti, pa čak i ulazno-izlazne jedinice. Također HWLOC prikuplja različite parametre kao što su informacije o priručnoj memoriji. Takav opširan ispis podataka korisniku može mnogo doprinijeti. Razvojni stručnjaci mogu donositi bolje i točnije odluke koje se svode na topologiju ovisno o podacima koji su prikupljeni alatom HWLOC budući da je njihov cilj što efikasnije iskoristiti sklopovlje, posebice u paralelnom programiranju i korištenju paralelnih aplikacija na podijeljenom sklopovlju, a administratori raznih sustava mogu pratiti stanje svojih servera. Nedvojbeno je da se HWLOC razvija velikom brzinom i njegova upotreba sve je veća. Može se izvoditi na širokom spektru računalnih sustava i računalnih sklopovlja. Kao što stoji na [2], operacijski sustavi na kojima HWLOC funkcionira su:

-Microsoft Windows

-Linux

-Solaris

-Aix

-Darwin / OS X

-FreeBSD

-NetBSD

-OSF/1

-HP-UX

3.1. Paralelno programiranje

Kod kompleksnijih programa važno je pravilno definirati algoritam, odnosno slijed naredbi koje su potrebne kako bi se programiranje izvelo na pravilan način. Kod sastavljanja algoritma, važno je znati kakvim se resursima raspolaže u računalu te kako sklopovlje predstaviti modelom. U tom slučaju javlja se potreba za alatima poput HWLOC-a jer arhitektura sklopovlja uvelike utječe na izvedbu raznih programa. Njegova uloga još je

važnija kada se radi o, primjerice, paralelnom programiranju. Paralelno programiranje je vrsta programiranja gdje se koriste posebne metode za razdvajanje algoritama ili problema na njegove osnovne dijelove i izvode se istovremeno na više računala ili centralnih procesorskih jedinica. Optimizacija performansi paralelnih aplikacija zahtjeva temeljno poznavanje arhitekture sklopovlja i cilj mnogih istraživanja i projekata je stvoriti alat koji će pomoći pri dostizanju tog cilja. Ono što je problematično kod prikupljanja informacija o topologiji je činjenica da je "izvlačenje" podataka jako težak proces jer se neki detalji o sklopovlju nalaze u datotekama koje su namijenjene korisniku za čitanje, a neki detalji su podijeljeni u više različitih datoteka pisanih strojnim jezikom.

3.2. Paralelne aplikacije na podijeljenom sklopovlju

Kako se danas sve više koriste višejezgreni procesori javlja se potreba da se pravilno i što bolje iskoristi što više njihove moći. Pravilnim načinom izvođenja programa moguće je smanjiti vrijeme koje je potrebno za pronalaženje rješenja. Na primjer, kada se obavljanje nekog zadatka koji je korisnik zadao vrši tako da se on seli s jedne jezgre procesora na drugu, njegovo izvođenje će biti sporije. Stoga bi se pojedini zadatak trebao obavljati unutar jedne jezgre kako bi se izbjeglo upravo to dijeljenje posla između više jezgri. Također, dijeljenje posla može uzrokovati varijacije u performansama ovisno o međudobnim lokacijama jezgri te u odnosu na ulazno-izlazne uređaje koji se koriste. Paralelne aplikacije često zahtjevaju komunikaciju i sinkronizaciju. Često se stoga više poslova koji su povezani trebaju odvijati na susjednim jezgrama kako bi se uspostavila što pravilnija komunikacija ili sinkronizacija među njima.

3.2.1. Model komunikacije dijeljenom memorijom

Model dijeljene memorije vjerojatno je najjednostavniji od tri modela. Zajednička memorija je poprilično efikasan način preko kojega procesi mogu razmijenjivati podatke. Različiti mehanizmi kao lokoti ili semafori koriste se kako bi se kontrolirao pristup dijeljenoj memoriji te kako bi se uklonilo "utkrivanje" među procesima jer procesi memoriji pristupaju asinkrono. Prednost ovakvog modela je što je njegov razvoj jednostavniji budući da svi procesi imaju jednaka prava pristupa zajedničkoj memoriji. Nedostatak bi bio taj što držanje podataka u blizini procesa koji rade na njima ponekad dovodi do zastoja kod pristupa memoriji, u prometu sabirnica i osvježavanjima priručne memorije kada više procesa koristi iste podatke.

3.2.2. Model komunikacije razmjenom poruka

U modelu razmjene poruka, paralelni procesi razmjenjuju podatke međusobnim slanjem i primanjem poruka. Slanje poruka može biti sinkrono, kada primatelj poruke mora biti spreman primiti poslanu poruku, te asinkrono, kada primatelj poruke ne mora biti spreman na primanje poslane poruke.

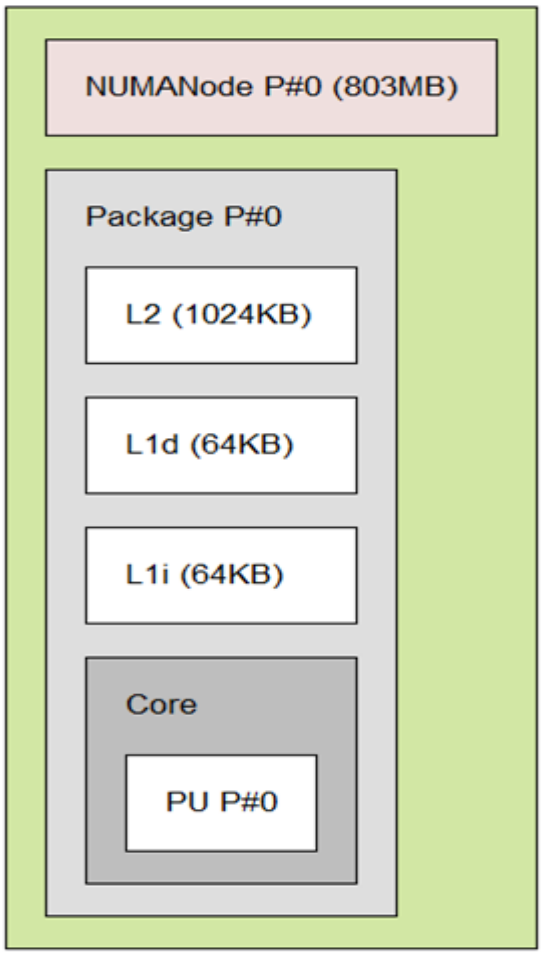
3.2.3. Model komunikacije podatkovnim paralelizmom

Model podatkovnog paralelizma se fokusira na izvođenje operacija na skupovima podataka koji su pravilno strukturirani (vektori ili polja). U ovom modelu skup zadataka se odvija nezavisno nad podskupovima podataka. Na arhitekturama dijeljene memorije, svi zadaci mogu imati pristup strukturi podataka kroz globalnu memoriju, dok na arhitekturama distribuirane memorije struktura podataka je razdvojena i nalazi se u dijelovima na lokalnoj memoriji svakog pojedinog zadatka – prema [3]

3.3. Instalacija i upotreba HWLOC-a – Microsoft Windows

HWLOC se može besplatno preuzeti sa službene stranice Open MPI tima (http://www.open-mpi.org/projects/hwloc/) i za njega nije potrebna klasična instalacija. Sve potrebne datoteke za njegov rad nalaze se u „.rar“ datoteci koju je samo potrebno raspakirati na željenu lokaciju. Nakon raspakiravanja, da bi se pokrenulo grafičko sučelje samog alata HWLOC i istražio izgled procesora koji se nalazi u računalnoj konfiguraciji računala koji koristi HWLOC, potrebno je u direktoriju „bin“ pokrenuti datoteku „Isotopo-win“.

Nakon pokretanja spomenute datoteke korisnik dobija grafički uvid u izgled i komponente koje se nalaze na njegovom procesoru računala. HWLOC pokrenut na računalu koje sadrži procesor „AMD Athlon(tm) Processor LE-1640 2.60GHz“ pružio je uistinu detaljan prikaz arhitekture procesora.



Slika 3.1: HWLOC-ov prikaz procesora AMD Athlon(tm) Processor LE-1640

Procesor AMD Athlon(tm) Processor LE-1640 sastoji se od jezgre procesora (Core PU P#0); dvije vrste L1 priručne memorije: L1i i L1d; L2 priručne memorije te NUMA memorije od 803MB kao što se može primijetiti na slici 3.1. Do sada je spomenuta L1 priručna memorija no ne i razlika između L1i i L1d priručne memorije. Naime, L1d (eng. *Level 1 data*) priručna memorija sadržava podatke, dok L1i (eng. *Level 1 instruction*) priručna memorija sadržava instrukcije.

3.4. Instalacija i upotreba HWLOC-a – Linux

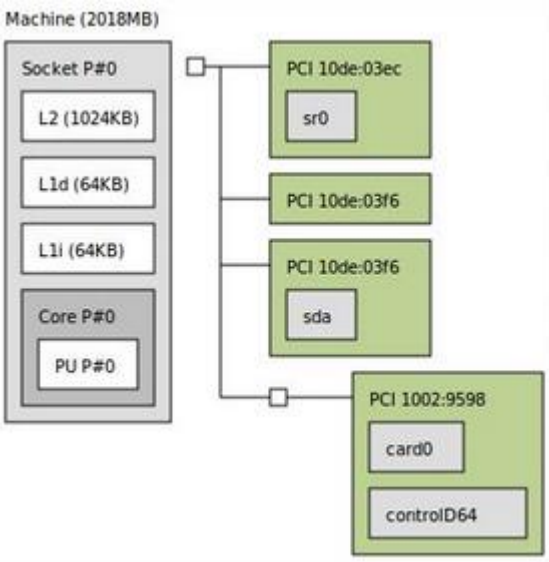
Svoje najbolje karakteristike ovaj alat pokazuje na operacijskom sustavu Linux. Za preuzimanje, instalaciju i testiranje HWLOC-a korištena je Ubuntu distribucija Linuxa. Samo nabavljanje i instalacija ovog alata na Linuxu moze se izvršiti na više načina. Jedan od tih načina je na stranici alata skinuti verziju za Linux te je instalirati ručno. Drugi način, onaj koji

je korišten u ovom radu je preko komandne linije. Od korisnika se traži da unese naredbu za dohvat paketa kako bi se on instalirao na računalo kao što stoji na [4].

U ovom slučaju ona glasi:

```
sudo apt-get install hwloc
```

Nakon unošenja te naredbe i pritiska na tipku ENTER, programski paket će se sam skinuti i instalirati, a korisnik će ga najjednostavnije moći nastaviti koristiti putem komandne linije i naredbi koje su različite za svaku posebnu operaciju koja se želi izvršiti s alatom. Ona najjednostavnija naredba je (napomenimo da se na Linuxu za unos naredbi za HWLOC zapravo koristi izraz `lstopo`, a ne `hwloc`) `lstopo`. Unošenjem te naredbe korisniku se prikazuje izgled njegove topologije, odnosno arhitekturu procesora te PCI utora koji se nalaze na matičnoj ploči.



Slika 3.2: HWLOC-ov prikaz arhitekture na operacijskom sustavu Linux

Nadalje, ukoliko se želi prikaz bez PCI utora i bridgeova unijet će se naredba:

`lstopo -no-io`

Slika 3.3: HWLOC-ov prikaz arhitekture bez PCI utora i bridgeova

Korisniku se pružaju i opširnije informacije o cijeloj arhitekturi unošenjem naredbe:

`lstopo -v`

```
ubuntu@ubuntu:~$ lstopo -v
Machine (P#0 local=2066576KB total=2066576KB DMIProductName=N61PB-M2S DMIProductVersion=6.0 DMIBoardVendor="BIOSTAR Group" DMIBoardName=N61PB-M2S DMIB
ardVersion=1.0 DMIBoardAssetTag= DMIChassisVendor="BIOSTAR Group" DMIChassisType=3 DMIChassisVersion=N61PB-M2S DMIChassisAssetTag=" " DMIBIOSVendor="P
oenix Technologies, LTD" DMIBIOSVersion="6.00 PG" DMIBIOSDate=06/16/2008 DMISysVendor="BIOSTAR Group" Backend=Linux OSName=Linux OSRelease=3.16.0-23-g
neric OSVersion="#31-Ubuntu SMP Tue Oct 21 18:00:35 UTC 2014" HostName=ubuntu Architecture=i686)
  Socket L#0 (P#0 CPUModel="AMD Athlon(tm) Processor LE-1640")
    L2Cache L#0 (size=1024KB linesize=64 ways=16)
      L1dCache L#0 (size=64KB linesize=64 ways=2)
        L1iCache L#0 (size=64KB linesize=64 ways=2)
          Core L#0 (P#0)
            PU L#0 (P#0)
  Bridge Host->PCI L#0 (P#0 buses=0000:[00-04])
    PCI 10de:03ec (P#96 busid=0000:00:06.0 class=0101(IDE) PCIVendor="NVIDIA Corporation" PCIDevice="MCP61 IDE") "NVIDIA Corporation MCP61 IDE"
      Block L#0 "sr0"
    PCI 10de:03f6 (P#128 busid=0000:00:08.0 class=0101(IDE) PCIVendor="NVIDIA Corporation" PCIDevice="MCP61 SATA Controller") "NVIDIA Corporation MCP6
 SATA Controller"
    PCI 10de:03f6 (P#129 busid=0000:00:08.1 class=0101(IDE) PCIVendor="NVIDIA Corporation" PCIDevice="MCP61 SATA Controller") "NVIDIA Corporation MCP6
 SATA Controller"
      Block L#1 "sda"
    Bridge PCI->PCI (P#144 busid=0000:00:09.0 id=10de:03e8 class=0604(PCI_B) buses=0000:[02-02] PCIVendor="NVIDIA Corporation" PCIDevice="MCP61 PCI Exp
ress bridge") "NVIDIA Corporation MCP61 PCI Express bridge"
      PCI 1002:9598 (P#8192 busid=0000:02:00.0 class=0300(VGA) PCIVendor="Advanced Micro Devices, Inc. [AMD/ATI]" PCIDevice="RV635 [Radeon HD 3650/375
/4570/4580]") "Advanced Micro Devices, Inc. [AMD/ATI] RV635 [Radeon HD 3650/3750/4570/4580]"
        GPU L#2 "card0"
        GPU L#3 "controlD64"
depth 0:        1 Machine (type #1)
 depth 1:       1 Socket (type #3)
  depth 2:      1 L2Cache (type #4)
   depth 3:     1 L1dCache (type #4)
    depth 4:    1 L1iCache (type #4)
     depth 5:   1 Core (type #5)
      depth 6:  1 PU (type #6)
Special depth -3:       2 Bridge (type #9)
Special depth -4:       4 PCI Device (type #10)
Special depth -5:       4 OS Device (type #11)
ubuntu@ubuntu:~$
```

Slika 3.4: Unošenje naredbe `lstopo -v`

Arihtektura se može prikazati i unutar same komandne linije. Ukoliko korisnik unese naredbu `lstopo –of console` dobit će tekstualni prikaz arhitekture:

```
ubuntu@ubuntu:~$ lstopo --of console
Machine (2018MB)
  Socket L#0 + L2 L#0 (1024KB) + L1d L#0 (64KB) +
L1i L#0 (64KB) + Core L#0 + PU L#0 (P#0)
  HostBridge L#0
    PCI 10de:03ec
      Block L#0 "sr0"
    PCI 10de:03f6
    PCI 10de:03f6
      Block L#1 "sda"
    PCIBridge
      PCI 1002:9598
        GPU L#2 "card0"
        GPU L#3 "controlD64"
```

Slika 3.5: Prikaz arhitekture unutar komandne linije

Spomenuta naredba `--of` koristi se kako bi korisnik odabrao u kojem formatu želi da se prikaže arhitektura. Podržani formati u ovom alatu su:

pretpostavljeni (kao na slici 3.2)

console – tekstualni prikaz u command lineu

txt – ASCII prikaz arhitekture u command lineu

fig – vektorska skica

pdf – arhitektura se sprema u pdf formatu

ps - arhitektura se sprema u postscript formatu

png - arhitektura se sprema u png formatu slike

svg – vektorska skica u xml formatu

xml – prezentacija arhitekture u EXtensible Markup Language (može biti sačuvana i kasnije korištena čak i na drugom računalu, naredbom za učitavanje `--input`)

Neke od dodatnih, manje važnih mogućnosti, bile bi mogućnost mijenjanja veličine fonta te debljine rubova naredbama `--fontsize <veličina>` i `--gridsize <veličina>` kao i mijenjanje boja pozadine i teksta naredbama `background` i `background2`, odnosno `text` i `text2`. Kao što je navedeno na [5], zanimljiva prednost ovog alata je mogućnost da

korisnik sam izradi neku virtualnu platformu tj. da sam dizajnira sintetičku arhitekturu nekog izmišljenog sklopovlja kako bi se korisnik na neki način mogao savjetovati bez da posjeduje određeno sklopovlje. Ukoliko, npr. korisnik želi replicirati sklopovlje na kakvom je testiran ovaj alat, on treba znati naredbu za generiranje virtualne platforme sa jednim socketom, jednom jezgrom (eng. *core*), jednom procesorskom jedinicom (eng. *pu – processing unit*) i tri priručne memorije.

Naredba za taj postupak bila bi:

```
lstopo –i „socket:1 core:1 pu:1 cache:1 cache:1 cache:1
```

Nadalje, ukoliko korisnik želi „stvoriti" sklopovlje koje se sastoji od, primjerice, 2 NUMA čvora, 2 socketa sa 2 jezgre na svakom, a u svakoj jezgri po 2 procesorske jedinice sa svojom priručnom memorijom, naredba koju će koristiti glasi:

```
lstopo –i „node:2 socket:2 core:2 pu:2 cache:1 cache:2
```

Slika 3.6: Izgled virtualne platforme koju je korisnik dizajnirao

4. CPU-Z

CPU-Z je alat osmišljen od strane tvrtke pod imenom CPUID. CPU-Z je jedan od najpoznatijih alata danas za identifikaciju i nadgledanje sklopovlja. Alat je besplatan i može ga se pronaći na službenoj stranici CPUID-a – [6] (http://www.cpuid.com/). Ovaj alat radi na operacijskim sustavima Microsoft Windows te Android. Dakle može se koristiti na raznim uređajima kao što su računala, mobiteli ili tableti.

4.1. Instalacija i upotreba – Microsoft Windows

Na već spomenutoj službenoj stranici CPUID-a na naslovnici može se preuzeti CPU-Z za dva različita operacijska sustava: Windows i Android. Navedeni alat prvo je testiran na operacijskom sustavu Windows. Nakon instalacije na željenu lokaciju koju bira korisnik, moguće je pokrenuti i u konačnici i koristiti ovaj alat. Pokretanjem CPU-Z-a možemo vidjeti sve podatke o:

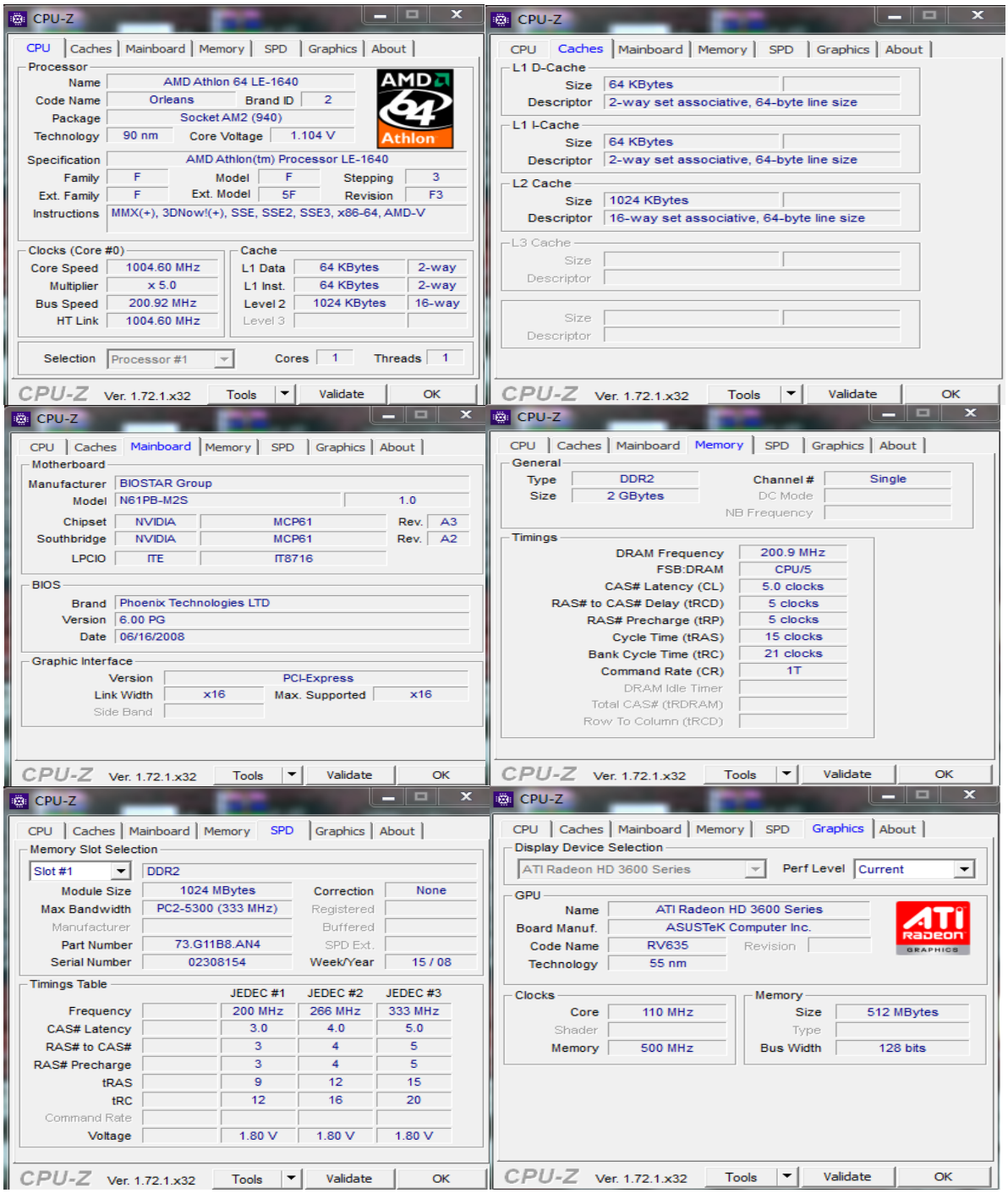
- procesoru: ime, tehnologija, napon, skupovi instrukcija, takt jezgre, priručna memorija...

- priručnoj memoriji: popis priručnih memorija, veličina, opis

- matičnoj ploči: proizvođač, model, podaci o BIOS-u, grafičko sučelje

- RAM memoriji (za svaki slot): tip, veličina, takt, frekvencija...

- grafičkoj kartici: ime, tehnologija, takt, memorija..

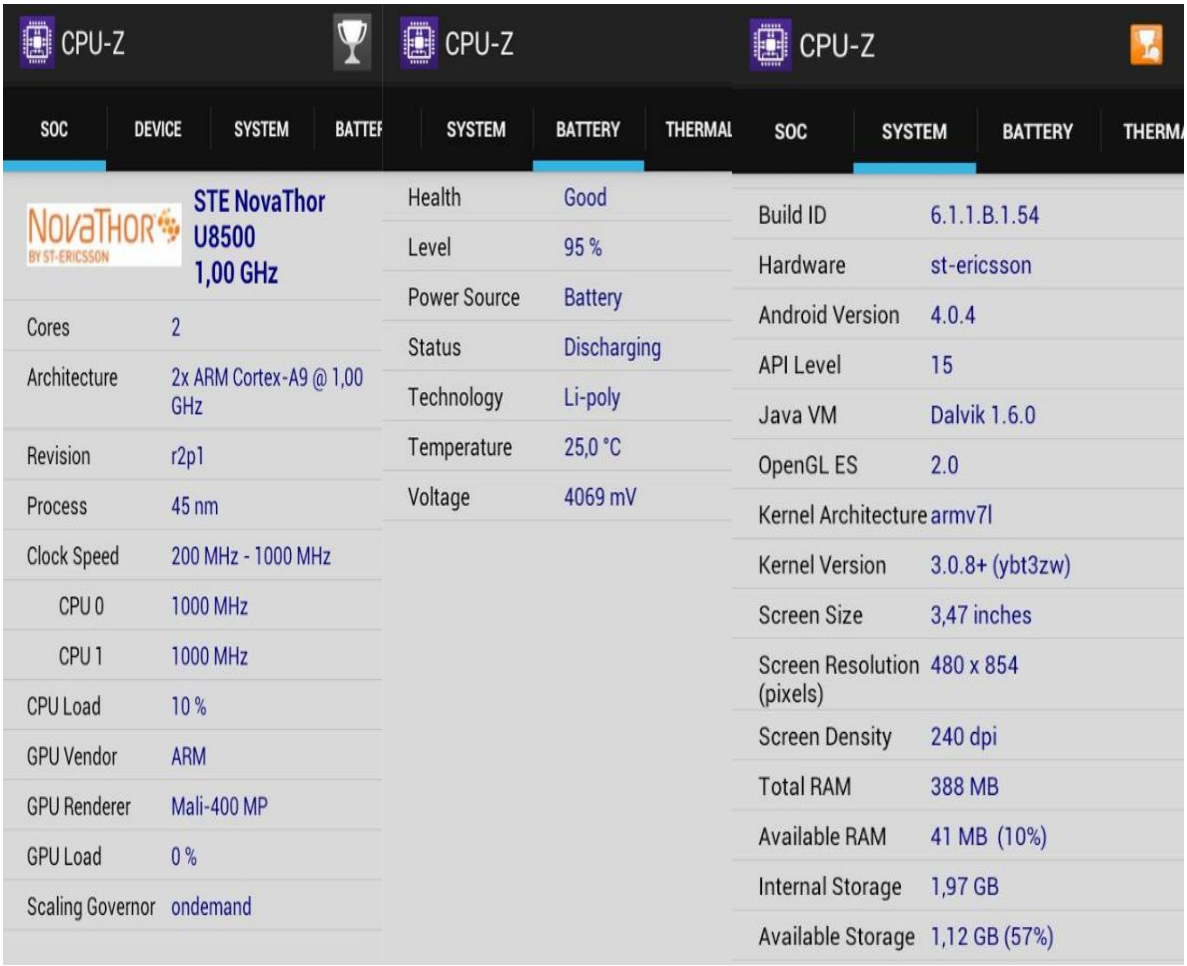


Slika 4.1: CPU-Z prikaz trenutne konfiguracije računala sa operacijskim sustavom Microsoft Windows

4.2. Instalacija i upotreba – Android

Alat CPU-Z postoji i za inspekciju sklopovlja na mobilnim uređajima i tabletima koji koriste operacijski sustav Android. Instalacija na spomenute uređaje vrši se putem trgovine

aplikacijama „Google play“. Korisnik na stranici CPUID-a može preko računala skinuti aplikaciju na svoj mobilni uređaj ili je može pomoću mobitela naći sam u online trgovini. Nakon pokretanja skinute aplikacije, ona će se sama instalirati i spremna je za korištenje. Korištenje je vrlo slično, poput verzije za računalo, a korisnik dobija informacije o procesoru, operacijskom sustavu, bateriji, termalnim i drugim senzorima (za naprednije mobilne uređaje).



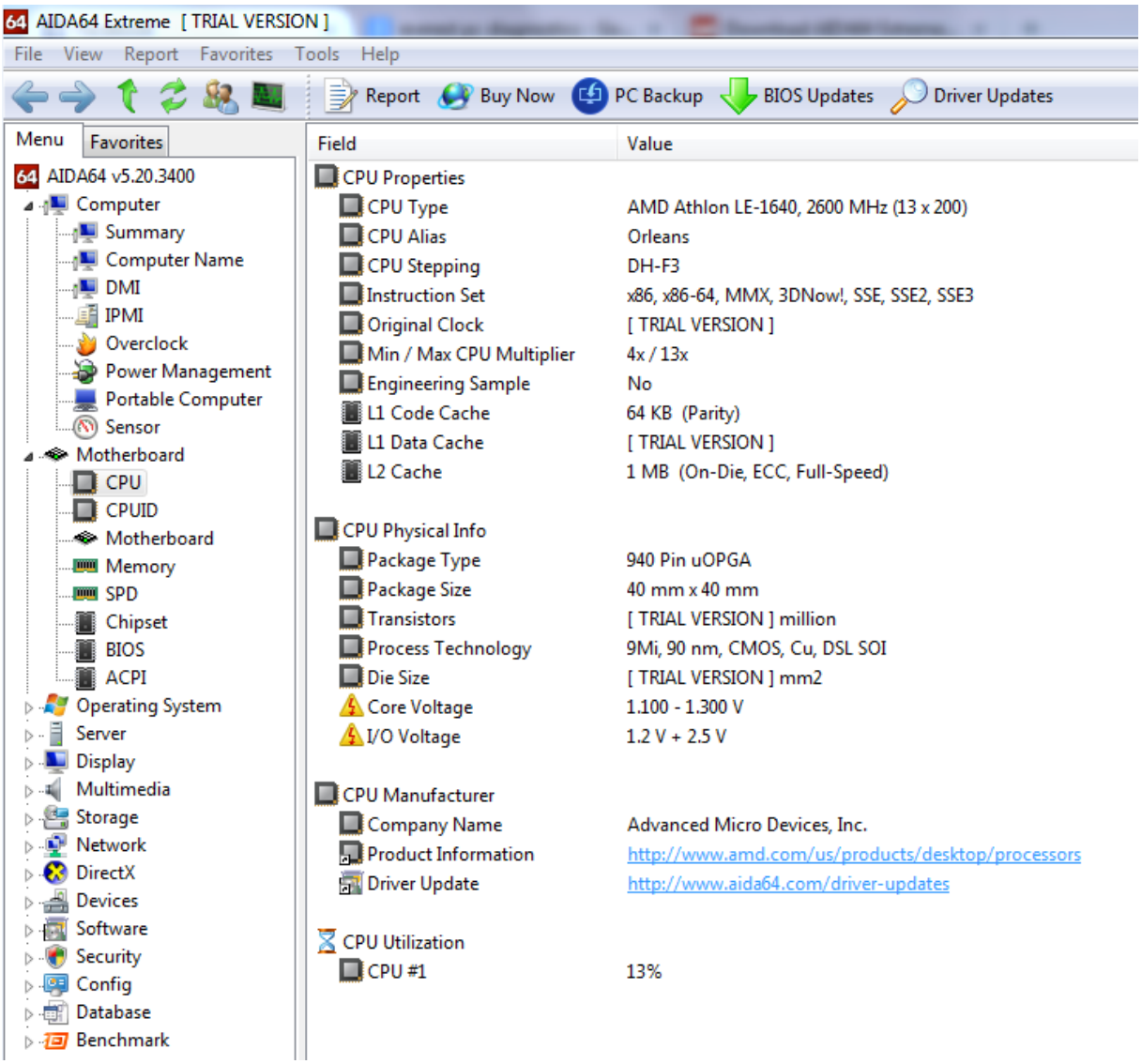
Slika 4.2: CPU-Z prikaz trenutne konfiguracije mobilnog uređaja (Android)

5. AIDA64

AIDA64, alat nekad poznatiji pod imenom Everest, prema [7], osmišljen je kao asistent korisniku kod instalacije, optimiziranja i otklanjanja poteškoća računala na način da pruža gotovo sve moguće informacije o istom. Pruža jasan uvid u sklopovlje i programsku podršku kao i detalje o operacijskom sustavu te njegovoj zaštiti i stabilnosti. Ovaj alat dostupan je korisnicima operacijskih sustava Microsoft Windows, Android te iOS što znači da se može podjednako kvalitetno koristiti i na računalima i na mobilnim uređajima te tabletima.

5.1. Instalacija i upotreba – Microsoft Windows

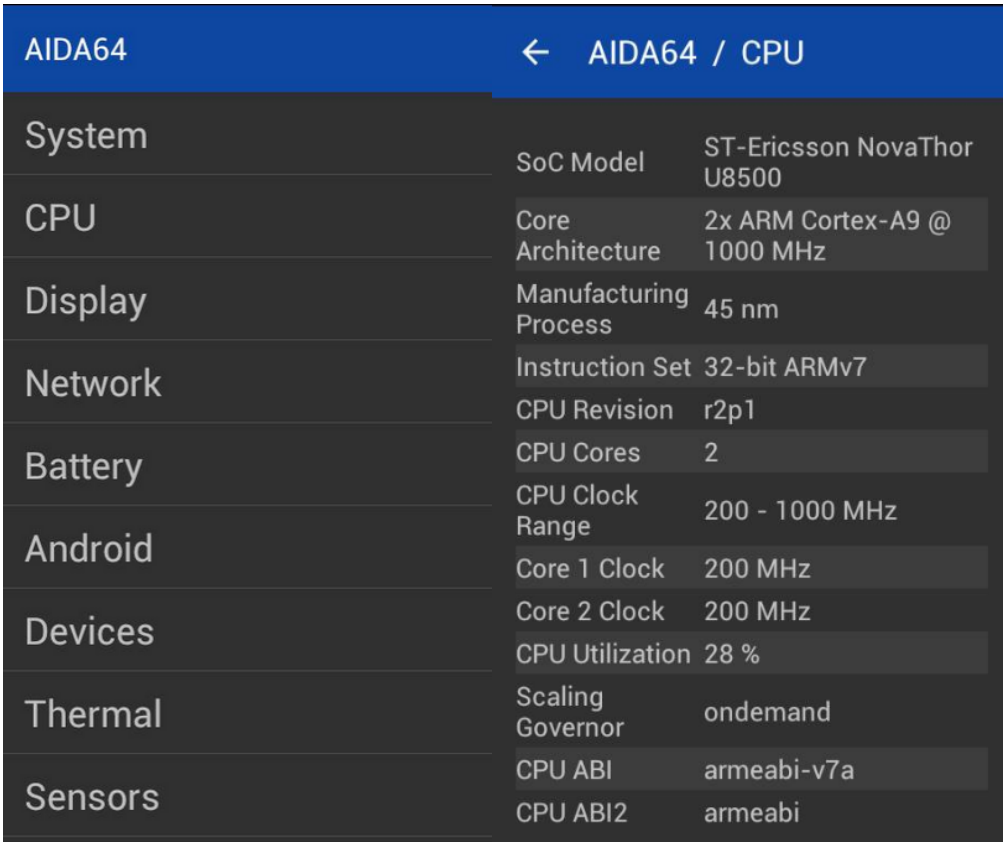
Ovaj alat se poput većine programa na sličan način preuzima i instalira na računalo. Korisnik prije instalacije može izabrati iz ponude više verzija te želi li skinuti odabranu verziju za instalaciju ili raspakiravanje iz arhive bez instalacije. Ukoliko korisnik želi koristiti ovaj alat besplatno, može ga skinuti besplatno, ali na period od 30 dana (tzv. „trial verzija“, odnosno verzija za isprobavanje kako bi korisnik odlučio želi li kupiti proizvod ili ne) što mu onemogućava neke od opcija koje bi imao kada bi kupio ovaj alat kao što su opcije mjerenja performansi svakog dijela sklopovlja posebno, kao i alati za inspekciju i nadgledanje računala povezanih u mrežu. Samo instaliranje i korištenje ovog alata vrlo je jednostavno, a izbornik pruža jako velik broj podataka i opcija za korisnika kao što je pregled općenitih podataka o računalu, pregled gotovo svih komponenti koje se nalaze na matičnoj ploči, informacije o operacijskom sustavu, pregled memorije, pregled ulazno-izlaznih jedinica i sl.



Slika 5.1: Izgled korisničkog sučelja alata AIDA64 za Microsoft Windows

5.2. Instalacija i upotreba – Android

AIDA64 za mobilne uređaje i tablete koji koriste operacijski sustav Android instalira se putem aplikacije online trgovine „Google play“. Instalacija se vrši automatski tako da korisnik nakon pokretanja skidanja ove aplikacije. Izgled verzije ovog alata za Andriod vrlo je sličan onome za računalo te korisniku nudi slične opcije i uvid u pojedine specifikacije.



Slika 5.2: Izgled glavnog izbornika (lijevo) i podataka za CPU (desno) aplikacije AIDA64 za Android

6. USPOREDBA ALATA

Kako bi se provela što opsežnija usporedba navedenih alata za inspekciju sklopovlja potrebno je u obzir uzeti što više parametara. Potrebno je istražiti pojedinosti i mogućnosti svakog od njih kako bi došlo do optimalnog odabira. Iako su gledajući izdaleka ovi alati jako slični, razlikuju se, primjerice po operacijskim sustavima na kojima rade, načinu prikaza pa čak i samom broj detalja koje prikazuju o pojedinom sklopovlju. Naravno, alati obrađeni u ovom radu nisu jedini alati tog tipa, no zbog svojih mogućnosti i popularnosti, neki su od najčešće korištenih.

Tablica 6.1: Usporedba HWLOC, CPU-Z i AIDA64

IME ALATA		HWLOC	CPU-Z	AIDA64
PREDNOSTI		-Prikaz sklopovlja s velikim brojem detalja -Velik broj mogućnosti -Korisnik ima mogućnost dizajniranja sklopovlja i uvid u raspoložive resurse	-Jednostavna instalacija -Jednostavno korištenje -Automatski prikaz sklopovlja -Mnogo detalja -Lako razumljiv svima	-Jednostavna instalacija -Pregledan prikaz sklopovlja -Pruža mogućnost testiranja sklopovlja pod raznim uvjetima
NEDOSTACI		-Nema jednake mogućnosti na svakom OS-u -Linux pruža više mogućnosti od ostalih OS-ova		-Potrebno platiti za punu verziju
OS	WINDOWS	✓	✓	✓
	LINUX	✓	–	–
	ANDROID	–	✓	✓

7. ZAKLJUČAK

Alati za inspekciju sklopovlja poput HWLOC-a, CPU-Z-a i AIDA64 veoma su jednostavni za upotrebu, a dostupni su gotovo svima. U ovom radu čitatelj je pobliže upoznat s njima, načinom na koji funkcioniraju te su pronađene njihove prednosti i nedostaci gledajući sva tri alata istovremeno. Njihovim korištenjem može se saznati mnogo o računalnom sklopovlju. Mogućnosti, primjerice, HWLOC-a sežu puno dalje od toga. Korisnik dobija mogućnost ne samo inspekcije svog sklopovlja, već i planiranja sklopovlja kakav je on zamislio da bi mogao biti. Alat HWLOC, za razliku od druga dva koja su spomenuta u ovom radu, više je namijenjen razvojnim stručnjacima te administratorima raznih računalnih sustava jer upravo oni pronalaze veću korist u njemu. Iskorištavanje svih njegovih mogućnosti i pravilno korištenje zahtjeva određenu dozu educiranosti o računalnom sklopovlju i programskoj podršci od strane korisnika. S druge strane, alati poput CPU-Z i AIDA64 namijenjeni su uglavnom „svakodnevnim“ korisnicima zbog svoje jednostavnosti i pristupačnosti. Nedvojbeno će se s vremenom i napretkom računalnog sklopovlja javiti potreba za unaprijeđivanjem ovih alata novim standardima i korisnicima preostaje vidjeti kako će se spomenuti alati nositi s time.

LITEATURA

[1] Martinović, G. (2016.) - materijali skolegija Operacijski sustavi, Elektrotehnički fakultet Osijek [online],
Dostupno na: https://loomen.carnet.hr/mod/folder/view.php?id=146406

[2] Službena stranica HWLOC projekta (19.06.2016.), Bloomington, Indiana, SAD [online]
Dostupno na: http://www.open-mpi.org/projects/hwloc/

[3] Kalafatić, Z., Hrkać, T. (2013./2014.) – višeprocesorski i paralelni sustavi, ZEMRIS, FER

[4] Nanni, D. (24.06.2016.) – osnivač stranice Xmodulo.com [online]
Dostupno na: http://xmodulo.com/identify-cpu-processor-architecture-linux.html

[5] Thibault, S. – (02.07.2012.) , Bordeaux, Francuska [online],
Dostupno na: https://www.open-mpi.org/projects/hwloc/tutorials/20120702-POA-hwloc-tutorial.pdf

[6] Službena stranica tvrtke CPUID (2016.) [online]
Dostupno na: http://cpuid.com/

[7] Norris, J. (26.04.2012.) – PC World [online]
Dostupno na: http://www.pcadvisor.co.uk/review/disk-tools-software/aida64-extreme-edition-review-3353628/

SAŽETAK

Tema završnog rada je „Izrada modela sklopovlja računalne platforme upotrebom alata HWLOC“. U izradi završnog rada korišteni su alati HWLOC, CPU-Z i AIDA64, a Lightshot i Paint su bili potrebni za uzimanje i uređivanje slika. U početku su programi testirani na Microsoft Windows operacijskom sustavu, a da bi se HWLOC mogao testirati sa svim mogućnostima bilo je potrebno upoznati se s radom na Linuxu i naredbama potrebnim za njegovo pravilno korištenje. Kako se bilo koji od ovih alata sa svojim mogućnostima testirao uzimale su se slike kao dokaz i prikaz rezultata. Kada su testiranja svih alata završena prikupljene podatke i slike bilo je potrebno objediniti u dokument i poštivati pritom pravila pisanja završnog rada.

Ključne riječi: HWLOC, sklopovlje, CPU-Z, AIDA64, Linux

ABSTRACT

The main goal of this thesis is „Making a hardware model of computer platform using HWLOC“. In the making of this paper I used tools: HWLOC, CPU-Z and AIDA64, while Lighshot and Paint were used to edit and save pictures. At the beginning, the tools were tested on Microsoft Windows operating system, but to get the best performances from HWLOC it was required to explore Linux and its commands. As the testings were made, pictures were taken as a proof of the results that were acquired. When the testings of all above mentioned tools were done, gathered data and pictures were summarized into one document, while the rules for making this paper were followed.

Key words: HWLOC, hardware, CPU-Z, AIDA64, Linux

ŽIVOTOPIS

Hrvoje Štadler rođen je 30.11.1992. godine u Osijeku. Školovanje je započeo Osnovnoj školi dr. Franje Tuđmana u Belom Manastiru. Nakon toga pohađao je Prvu srednju školu Beli Manastir, gdje je stekao zvanje tehničara za računalstvo. Trenutno je student 3. godine stručnog studija informatike na Fakultetu elektrotehnike računarstva i informacijskih tehnologija u Osijeku. Od stranih jezika, služi se engleskim. Posjeduje znanje rada na računalu u različitim programskim jezicima te u korištenju raznih alata (Microsoft office, Visual studio, AutoCAD, Matlab, Web programiranje i sl.).

Potpis: _____