

Šahovski program Umko za Android

Damjan Casar, Borko Bošković, Janez Brest

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko

Inštitut za računalništvo

Smetanova ulica 17, 2000 Maribor, Slovenija

E-pošta: damjan.casar@uni-mb.si

Umko Chess Engine on Android

In the paper we present the process of porting the best Slovenian open source chess program written in C++ programming language on the Android platform. This was done with the use of default native development kit available to download for Android called NDK version r5c. The application was ported using the g++ compiler for ARM architecture that is a part of the NDK. For the results we used an Android application called Chess for Android and the compiled version of Umko which was imported into the mentioned application. We then tested the application on an Android based tablet PC including it in a tournament with other chess engines. The conclusions are that the finished standing between different engines was approximately the same as on a regular PC and the ported program, running on the mobile device, still remains a strong chess player.

1 Uvod

V članku predstavljamo postopek prenosa šahovskega programa Umko, napisanega v programskem jeziku C++, na operacijski sistem Android. Slednji ni neposredno kompatibilen s programi, napisanimi v programskem jeziku C++.

Pri prenosu smo uporabili orodje Native Development Kit (NDK) [4] za prevajanje izvorne kode napisane v programskem jeziku "C" ali "C++" za arhitekturo ARM [10]. To arhitekturo uporabljajo pametni telefoni in tablični računalniki z nameščenim operacijskim sistemom Android. Po prevajanju smo izvedljivi program prenesli na tablični računalnik. S pomočjo programa "Chess for Android" [7], ki je grafični uporabniški vmesnik (angl. GUI), smo naložili program. Po nalaganju smo program preizkusili v turnirju z drugimi svetovno znanimi šahovskimi programi.

V drugem poglavju opisujemo operacijski sistem Android in arhitekturo ARM [10, 11]. Nato v tretjem poglavju na kratko opišemo šahovski program Umko [9]. Prenos programa na Android [1] je opisan v četrtem poglavju. V petem poglavju opišemo dobljene rezultate, ter nato v šestem podamo kratki zaključek.



Slika 1: Struktura Android operacijskega sistema po nivojih [8].

2 Android in arhitektura ARM

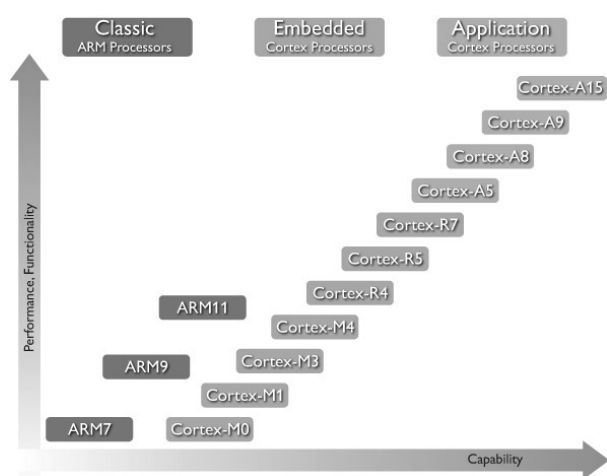
Operacijski sistem Android je odprto-kodni sistem. Podoben je ostalim operacijskim sistemom in uporablja hierarhično strukturo, ki jo prikazuje slika 1. Struktura vsebuje 4 nivoje: aplikacijski nivo, nivo za upravljanje z aplikacijami, sistemski nivo in nivo realno-časovnega jedra Linux. V aplikacijskem nivoju so aplikacije, ki so nameščene na sistemu. Drugi nivo sestavljajo osnovne aplikacije oz. upravljalniki aplikacij z vsemi dovoljenji za dostop do API-ja. Tretji nivo sestavljajo knjižnice, ki so napisane v programskem jeziku C/C++. Zadnji, četrti, nivo je jedro Linux, razvito v programskem jeziku C, ki predstavlja nivo med strojno opremo in sistemsko programsko opremo in nudi sistemske storitve, kot so:

- varnost,
- upravljanje s pomnilnikom,
- upravljanje z datotečnimi sistemi,
- upravljanje s procesi,
- mrežno skupino,
- gonilniki itd.

Android teče na ARM arhitekturi. Ta arhitektura je 32 bitna in vsebuje zmanjšani nabor inštrukcij (RISC). Je najbolj uporabljeni nabor ukazov (ang. Instruction set architecture-ISA), ki ima tudi naslednje funkcionalnosti:

- inštrukcije, ki kombinirajo premik z aritmetično ali logično operacijo,
- samodejno povečevanje in zmanjševanje načinov naslavljanja za optimizacijo zank,
- nalaganje in shranjevanje več inštrukcij za povečanje prepustnosti podatkov in
- pogojno izvajanje skoraj vseh inštrukcij za povečanje prepustnosti izvajanja.

Relativna preprostost ARM procesorjev jih naredi primerne za aplikacije z majhno porabo. Cenovna ugodnost, majhna poraba, majhni mikrokrmilniki in mikroprocesorji so jih naredili prevladujoče na področju mobilne in vgradne elektronike. Tako je bilo leta 2005 več kot 98% od neke ene milijarde mobilnih telefonov opremljenih z ARM procesorji [10].



Slika 2: Primerjava zmogljivosti in sposobnosti različnih ARM procesorjev [11].

Slika 2 prikazuje graf napredovanja oz. razvoja procesorjev z arhitekturo ARM [11]. Na sliki so različni tipi procesorjev razporejeni glede na zmogljivost in funkcionalnost v odvisnosti s sposobnostjo. Omenjeni so t.i.:

- Klasični procesorji ARM in so idealni za organizacije, ki želijo prenesti ustaljene tehnologije v nove aplikacije. Ti procesorji prinašajo številne funkcionalnosti, odlično učinkovitost in zmogljivost za cenovno ugodne rešitve.
- Vgradni procesorji ARM so narejeni za zelo raznolika tržišča. Serija procesorjev Cortex-M je bila razvita za mikrokrmilnike, kjer se uporabljajo procesorji z naslednjimi lastnostmi: hitri, visoko deterministični, imajo možnost upravljanja s prekinitvami, zelo majhno število vrat in maksimalno varčni. Serija procesorjev Cortex-R je bila razvita za vgradne realno-časovne aplikacije, kjer je potreba po varčnih in možnosti upravljanja s prekinitvami uravnovešena z izjemno zmogljivostjo in dobro združljivostjo z že obstoječimi platformami.

- Aplikacijski procesorji ARM prinašajo izjemno zmogljivost do 2 GHz tipične frekvence v naprednih procesnih vozliščih. Omogočajo naslednjo generacijo mobilnih internetnih naprav. Procesorji, ki so na razpolago, so eno-jederni ali več-jederni in lahko vsebujejo do štiri procesne enote. Uporabljajo se v večini multimedijskih prenosnih naprav, kot so: pametni telefoni, netbooki, e-bralniki, tablični računalniki, digitalna TV, domači mrežni prehodi (gateways) in druge.

3 Šahovski program Umko

Slovenski šahovski odprtokodni program Umko, katerega avtor je Borko Bošković, je napisan v programskem jeziku C++. Program je objavljen na spletnem sistemu SourceForge [9] pod GPLv3 licenco. Umko združuje dobre koncepte odprto-kodnih programov v eno celoto. Tako na najbolj ažurnih mednarodnih lestvicah Umko dosega okoli 2900 rating točk [3, 12]. Na omenjenih lestvicah se program uvršča med deset najboljših odprtokodnih programov. Slednje dosega s pomočjo uporabe otvoritvenih knjižnic programa Polyglot in podatkovnih baz končnic programa Gaviota. Program nima lastnega grafičnega vmesnika, je konzolna aplikacija, ki komunicira z različnimi uporabniškimi vmesniki. Program omogoča več variantno iskanje, kar pomeni, da izbrane variante preiskuje natančneje in omogoča tudi paralelno preiskovanje oz. izkoriščanje več procesorjev oz. jeder hkrati. V igrah zna izkoriščati t.i. “pondering mode”, kar pomeni, da razmišlja tudi kadar ni na potezi.

4 Prenos programa na Android

Za prenos programa na operacijski sistem Android smo uporabili že prej omenjeni NDK (Native Development Kit), ki smo ga prenesli s spletne strani [1]. To razvojno okolje vsebuje prevajalnik za Android (Cross compiler), kateri teče na operacijskem sistemu Linux. Prenesena verzija je bila “android-ndk-r5c” za operacijski sistem Linux (x86). Dobljeni paket ima dve možnosti prevajanja oz. izvajanja programov napisanih v programskem jeziku C/C++.

Prva možnost je uporaba t.i. vmesnika JNI (Java Native Interface). Slaba lastnost te možnosti je hitrost, ker se klici izvajajo preko vmesnika in se izvajajo počasneje, kar pomeni dodatno izgubo pri časovno omejenem računanju nadaljnjih potez.

Druga možnost, katero smo uporabili, je t.i. “native” prevajanje, kjer direktno prevajamo izvorno kodo napisano v programskem jeziku C/C++. Prevajanje smo izvedli po naslednjih korakih:

1. Določitev poti do NDK-ja s konzolnim ukazom:

```
export NDK=/home/user/android-ndk-r5c
```
2. Izbira med različnimi Android API-ji oz. verzijami. Mi smo izbrali android-8 oz. Android 2.1 (Froyo):

```
SYSROOT=$NDK/platforms/android-8/  
arch-arm
```

3. Ustvarjanje prilagojenega t.i. "toolchain-a", ki vsebuje vse potrebne datoteke za prevajanje:

```
$NDK/build/tools/make-standalone-  
toolchain.sh --platform=android-8  
--install-dir=/tmp/my-android-  
toolchain
```

Omenjeni ukaz kliče skripto, ki ustvari "toolchain" za Android 2.1 v namestitveni direktorij /tmp/my-android-toolchain.

4. Za direktno uporabo prej ustvarjenega razvojnega okolja, dodamo spremenljivki okolja *PATH* pot do prej ustvarjenih orodij:

```
export PATH=/tmp/my-android-toolchain/  
bin:$PATH
```

5. Določeni deli kode programa Umko so napisani v zbirnem jeziku, kar omogoča programu, da uporablja določene inštrukcije Intelovih procesorjev, ki jih sicer ne bi mogli uporabiti le z uporabo prevajalnika. Del teh inštrukcij je tudi nabor inštrukcij novejših arhitekture procesorjev SSE4. Te odseke, ki so odvisni od arhitekture procesorja, smo morali prilagoditi ARM arhitekturi.

6. S pomočjo predprocesorskih ukazov smo v času prevajanja ugotovili ciljno platformo. V primeru Android smo:

- velikost transpozicijske tabele nastavili na 12 MB,
- predpomnilnik baze končnic programa Gaviota nastavili na 4 MB.

7. Program Umko ima tudi datoteko za prevajanje (Makefile), katero smo prilagodili prevajanju za platformo Android:

- nastavljena sta prevajalnika GCC in G++ za arhitekturo ARM (arm-linux-androideabi-g++, arm-linux-androideabi-gcc),
- nastavljen je povezovalnik LD (arm-linux-androideabi-ld),
- nastavljen je program za arhiviranje AR (arm-linux-androideabi-ar),
- posodobljena so navodila tako, da vključujejo možnost prevajanja za operacijski sistem Android.

Program Umko nima grafičnega vmesnika, ampak je konzolna aplikacija. Grafični vmesnik (Chess for Android) zažene naš program kot zunanji proces in ga upravlja s pomočjo standardnih vhodno-izhodnih tokov in UCI protokola. Zato ni bilo potrebe po pisanju programa v programskem jeziku Java. Tako smo lahko uporabili originalno izvorno kodo napisano v programskem jeziku C++ in prilagodili določene dele kode napisane v zbirnem jeziku. Po prevajanju in

povezovanju smo dobili dinamično povezan izvedljiv program za operacijski sistem Android. Dinamično povezani program deluje le v verziji Androida, za katerega je bil preveden, zato smo dodali statično prevajanje z uporabo dodatne zastavice "LDLFLAGS += -static". Tako smo dobili program, ki teče na različnih verzijah operacijskega sistema Android.

5 Predstavitev rezultatov

Program smo preizkusili s pomočjo turnirja, ki je bil odigran na tabličnem računalniku. Uporabljen tablični računalnik je vseboval procesor ARM Cortex-A8 (TI OMAP 3630) 1 GHz, 256 MB RAM, 8 GB notranjega pomnilnika in operacijski sistem Android 2.2 (Froyo).

Po prevajanju smo izvedljivi program prenesli na spominsko kartico omenjenega tabličnega računalnika. Grafični vmesnik Chess for Android, ki podpira UCI protokol [5], je na voljo na spletišču za prenos Android aplikacij (Android market) [13]. Sam program je razvit kot univerzalni šahovski vmesnik in omogoča vključitev kateregakoli šahovskega programa, ki podpira UCI vmesnik. Šahovski program Umko namestimo v aplikacijo Chess for Android [6], ga omogočimo in že lahko igramo proti našemu programu.

Da bi ugotovili moč programa, smo ga preizkusili v turnirju šahovskih programov. Dobljene rezultate iger prikazuje tabela 1. Prvi stolpec prikazuje končni vrstni red programov, drugi ime programa, tretji dosežene rating točke, četrti in peti stolpec določata 95% interval zaupanja, šesti število odigranih iger, sedmi uspešnost v vseh igrah, osmi povprečni rating nasprotnikov in deveti stolpec odstotek remijev.

V turnir so bili vključeni svetovno znani šahovski programi. Njihove rating točke so bile izračunane s pomočjo programa Bayeselo [15], ki uporablja algoritem Minorization–Maximization [14] in ni imel nobenega referenčne informacije o realnem ratingu. Iz tega stališča smo pri izračunu dodali odmik, ki je znašal 2000 rating točk. Tako smo dobili relativne rezultate, ki so prikazani v tabeli 1. Dobljene rezultate smo primerjali z realnimi ažurnimi podatki, uporabljenih šahovskih programov z lestvice CCRL (Computer Chess Rating Lists) [3]. Podatke z lestvice CCRL prikazuje tabela 2.

Rezultati, predstavljeni v tabeli 1 in 2, so prikazani tudi na sliki 3. Na sliki je razviden enakomerni razmik med razlikami šahovskih programov turnirja na tabličnem računalniku in rezultati šahovskih programov na CCRL lestvici.

6 Zaključek

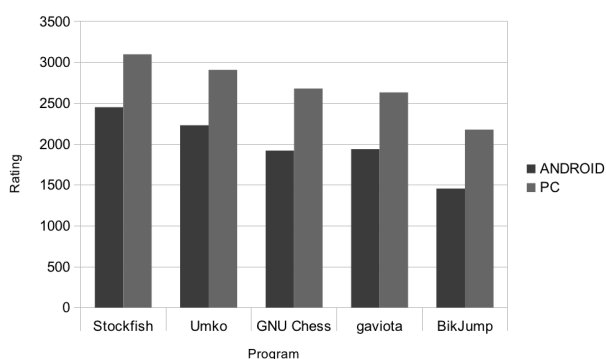
V članku je bil predstavljen prenos programa Umko napisan v programskem jeziku C++ na operacijski sistem Android. Čeprav je Android načrtovan tako, da se aplikacije napisane v Javi smo naš program prevedli s prevajalnikom za programski jezik C++. Android uporablja jedro Linux, torej uporablja tudi knjižnice za zagon aplikacij in programov napisanih v programskem jeziku C/C++. Tako smo s pomočjo Native Development Kit, izvorno kodo programa Umko prevedli za Android.

Tabela 1: Rezultati turnirja šahovskih programov na Androidu.

Rang	Ime	Rating	+	-	Igre	Rezultat	Naspr.	Remi
1	Stockfish 2.1	2452	161	161	40	95%	1887	5%
2	Umko 1.2 arm	2232	126	126	40	74%	1942	8%
3	gaviota v0.83	1939	126	126	40	41%	2015	8%
4	GNU Chess 5.07.153.3b-32	1921	128	128	40	40%	2020	5%
5	Chess for Android	1455	232	232	40	0%	2136	0%

Tabela 2: Tabela realnih rating točk po CCRL lestvici.

Rang	Ime	Rating	+	-	Igre	Rezultat	Naspr.	Remi
1	Stockfish 2.1.1 x64	3100	31	31	312	57,5%	3052	50,3%
2	Umko 1.0 x64	2907	29	29	368	52,0%	2892,6	39,9%
3	GNU Chess 5.07.153 x64	2679	34	33	307	61,9%	2595,3	30,6%
4	gaviota v0.80 x64	2633	32	31	344	51,3%	2623,4	27,0%
5	BikJump 2.01	2177	19	19	931	49,2%	2183,6	28,4%



Slika 3: Graf primerjave šahovskih programov po rating točkah na ARM procesorju in na Intel x64.

Sledila je vključitev programa v grafični šahovski vmesnik (Chess for Android). Na koncu smo program preizkusili v turnirju s 4 svetovno znanimi šahovskimi programi na tabličnem računalniku. Iz dobljenih rezultatov je razvidno, da je razlika rezultatov med šahovskimi programi glede na CCRL lestvico približno enaka. Na osnovi rezultatov lahko zaključimo, da je bil prenos programa uspešen in da je program kljub temu, da smo mu določene segmente kode spremenili, še vedno močan šahovski igralec.

Literatura

- [1] Yonghong Wu, Jianchao Luo, Lei Luo: Porting mobile web application engine to the Android platform, 10th International Conference on Computer and Information Technology (CIT 2010)
- [2] Dan Galphin: Coding to the Metal Developing with the Android NDK, Developer Day presentation, (Google 2010)
- [3] CCRL, http://computerchess.org.uk/ccrl/4040/rating_list_all.html
- [4] NDK, <http://developer.android.com/sdk/ndk/index.html>
- [5] UCI, <http://www.aartbik.com/MISC/eng.html>

[6] Chess for Android, <http://www.aartbik.com/MISC/amenu.html#chess>

[7] BikJump, <http://www.aartbik.com/MISC/chess.html>

[8] Architecture diagram of Android, http://en.wikipedia.org/wiki/File:Diagram_android.png

[9] Umko, dostopno na: <http://umko.sourceforge.net/>, http://labraj/index.php/Šahovski_program_Umko

[10] ARM, http://en.wikipedia.org/wiki/ARM_architecture

[11] ARM arhitektura, <http://www.arm.com/products/processors/index.php>

[12] CEGT, http://www.husvankempen.de/nunn/40_40%20Rating%20List/40_40%20All%20Versions/rangliste.html

[13] Android market, <https://market.android.com/>

[14] Hunter DR: (2004) MM algorithms for generalized Bradley-Terry models. *Ann Stat* 32(1):384-406

[15] Bayeselo, dostopen na: <http://remi.coulom.free.fr/Bayesian-Elo/>