



Povzetek projekta Po kreativni poti do znanja 2017 – 2020, 2. odpiranje, za namen objave in predstavitve na spletni strani sklada

1. Polni naslov projekta: Pospeševalnik navideznega stroja za vgrajene sisteme.

- V katero področje na prvi klasiifikacijski ravni KLASIUS-P-16 se uvršča projekt glede na vsebinsko zasnovo (neustrezno področje izbrišite):

06 - Informacijske in komunikacijske tehnologije (IKT)

2. V sodelovanju z: (navede se univerza oz. samostojni visokošolski zavod, ki je prijavil projekt in članica, ki je nosilka projekta ter partner/ja – podjetje/ji oz. organizacija, ki je/sta bilo/i vključeno/i v projekt)

Univerzo v Ljubljani (Fakulteta za računalništvo in informatiko) in
iSYSTEM Labs, d.o.o.

3. Besedilo:

- Opredelite problem, ki se je razreševal tekom izvajanja projekta

Prilagajanje programske opreme za mikrokrmilnike postaja vedno težje obvladljivo. Z možnostjo poganjanja uporabniških programov na nivoju emulatorja lahko uporabniki sami dodajajo podporo za mikrokrmilnike. Potrebno pa je preprečiti, da bi napake v uporabniških programih povzročile nepravilno delovanje ali celo okvaro emulatorja. V takih primerih je ugodno imeti na voljo softversko emulacijo, kot je npr. navidezni stroj EVE (Embedded Virtual Engine), ki pa je za nekatere naloge prepočasen, saj deluje kot interpreter (kodo ARMv6 prebere iz pomnilnika, jo sproti dekodira in izvaja). Po potrebi preveri še naslove pomnilniških dostopov in klicane funkcije. Ta postopek v povprečju traja 100 urinih period na ukaz.

Cilj projekta je bila implementacija pospeševalnika za navidezni stroj EVE. Pospeševalnik ima v principu enako funkcionalnost kot EVE, le precej hitrejši je. Ne deluje kot interpreter, ampak pred izvajanjem vnaprej prevede ukaze iz nabora ARMv6 v ukaze procesorja, na katerem izvajalni stroj teče. Pospeševalnik v nasprotju z navideznim strojem ukaze dekodira le v fazi prevajanja (translator), ne pa sproti med izvajanjem, zaradi česar deluje precej hitreje. Programsko kodo za ARMv6 translator pretvori v kodo C++, ki jo prevajalnik prevede v strojno kodo na gostiteljevi platformi.

- Opišite potek reševanja problema oz. kratek povzetek projekta

Napisali smo dva programa (projekta v Visual Studiu):

- translator, ki prevede obstoječo ARMv6 kodo iz binarne datoteke (*.bin) v kodo v jeziku C++, ter
- pospeševalnik, ki predstavlja model procesorja (nabor ARMv6), na katerem se izvaja koda, ki jo je generalni translator. Ta koda se najprej prevede in poveže z modelom procesorja v izvršilno kodo, ob zagonu pa pospeševalnik izvede ustrezno prevedeno kodo.

Poseben problem je bil reševanje problema odvisnosti ciljev skokov od podatkov. Kljub temu, da je nivoju translacije možno ugotoviti za mnoge primere, kje v programskem segmentu so ukazi in kje podatki (konstantni), pa to ni vedno izvedljivo v času prevajanja. Namreč, določeni skoki so odvisni

od trenutnega stanja nekega registra, to pa je lahko odvisno od podatkov, ki jih med prevajanjem ne poznamo. V takih primerih niti translator ne more ugotoviti, kje vse je koda. To je povsem jasno šele v času izvajanja. Zasnovali in implementirali smo algoritem, ki je zmožen obravnave takih primerov.

Testiranje je potekalo na več nivojih: po eni strani so implementatorji ukazov testirajo njihovo delovanje, poleg tega pa eden od študentov izvaja neodvisne teste na osnovi svojega (neodvisnega) vpogleda v delovanje ARMovih ukazov. Oboji so namenjeni preverjanju pravilnosti delovanja pospeševalnika. Testi izvrševanja so avtomatizirani - naredili smo orodje exeTest.py, ki gre samodejno čez celoten proces, ki zna biti relativno zapleten (najprej smo ga izvajali ročno): prevajanja, translacije ARMove kode, prevajanja generirane C++ kode z modelom procesorja, izvajanja pospeševalnika ter na koncu primerjave stanja procesorja s stanjem EVE po vsakem ukazu.

Testirali smo tudi podporne knjižnice in disassembler, ter preučili možnosti razhroščevanja s tehnologijo ARM Coresight, ki omogoča razhroščevanje in tracing ARM kode.

Hitrost delovanja pospeševalnika smo testirali na nekaj različnih programih, ki uporabljajo tipične programske konstrukte (vgnezdene zanke, klici funkcij, rekurzija).

Izvedli smo tudi testiranje klicev 'native' funkcij ('native call'), ki jih kličemo iz ARMove kode, so pa zunanje glede na ARM ('native' na računalniku gostitelju, na katerem poganjamo pospeševalnik).

- Navedite in opišite rezultate projekta ter njihov doprinos k družbeni koristnosti

Glavni rezultat projekta je pospeševalnik za navidezni stroj EVE, ki izvaja nabor ukazov ARMv6. Testiranje hitrosti je pokazalo, da je pospeševalnik v povprečju približno 20x hitrejši od stroja EVE, seveda pa je odvisno od konkretnega programa in pogostosti pojavljanja različnih tipov ukazov v njem. Poleg pospeševalnika smo v okviru projekta izvedli tudi teste knjižnic za EVE, teste disassemblerja ter preučili različne možnosti razhroščevanja.

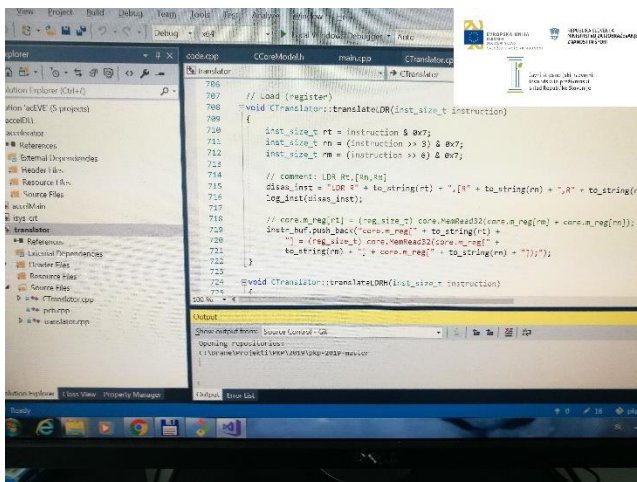
Podjetje iSYSTEM Labs izdeluje elektronsko opremo za nekatera področja, na katerih zelo pomembno vlogo igra vidik varnosti. Ta oprema mora delovati zelo zanesljivo, saj morebitne napake lahko vodijo do resnih posledic. Boljši izdelki podjetja iSYSTEM Labs, ki jih uporabljajo številni svetovni proizvajalci, bodo prispevali k razvoju zanesljivejših vgrajenih sistemov na področju avtomobilske in letalske industrije ter medicinske opreme.

Pedagoška mentorja bosta na Fakulteti za računalništvo in informatiko in na Fakulteti za elektrotehniko vključila znanja in izkušnje s projekta pri svojem pedagoškem delu, ki zadeva ravno procesorje, vgrajene sisteme, programiranje in sistemsko programsko opremo. Od izmenjave znanj in izkušenj med izobraževalno sfero in gospodarstvom preko skupnega dela na projektu bosta imeli obe strani od tega koristi, ne le kratkoročno, ampak tudi na daljši rok. Po eni strani dobijo pedagogi na univerzi boljšo predstavo o tem, kako poteka delo v industriji in kaj industrija potrebuje. Na osnovi tega bodo lažje vključili v pedagoški proces tudi relevantne principe iz prakse in gospodarstva. Na ta način bodo študenti tudi bolje opremljeni pri iskanju dela.

4. Priloge:

- Slikovno gradivo: Priložite vsaj dve sliki npr. sliko končnega produkta, sliko študentov pri delu na projektu, sliko s sestankov ipd. Pri pošiljanju slik bodite pozorni, v kolikor gre za končni produkt, da bo zadoščeno zahtevam glede informiranja in obveščanja (ustrezni logotipi itd.).

Aplikacija - translatore in pospeševalnik:



Mentorji na projektu:

