

Karakterizacija vremenskih parametara RT Linuks baziranog ugrađenog računarskog sistema

Boris Gluvić, Branko Dokić, Željko Ivanović

Elektrotehnički fakultet u Banjaluci

Univerzitet u Banjaluci

Banja Luka, Republika Srpska, Bosna i Hercegovina

gluvicboris@gmail.com, bdokic@etfbl.net, zeljko@etfbl.net

Sažetak—U ovom radu predstavljena je upotreba operativnog sistema RT Linuks u ugrađenim računarskim sistemima. Opisani su osnovni moduli i karakteristike operativnog sistema RT Linuks. Prikazane su vremenske karakteristike ovog operativnog sistema, način njegovog pokretanja na razvojnom okruženju *Pandaboard ES*, te su navedene prednosti i nedostaci ugrađenih računarskih sistema koji koriste operativne sisteme za rad u realnom vremenu.

Ključne riječi—Ugrađeni računarski sistem; Mikrokontroler; Operativni sistem;

I. UVOD

Ugrađeni računarski sistemi su sistemi sa konkretnom namjenom, za razliku od računarskih sistema opšte namjene. Funkcionalnost ugrađenog računarskog sistema je direktno povezana sa sistemom i okruženjem u kojem se on nalazi. Ovi računarski sistemi nalaze široku primjenu u uređajima potrošačke elektronike, automobilske industriji, vojnoj industriji, kućnoj automatizaciji, oblasti *IoT (Internet of Things)* uređaja i funkcionalno-kritičnim sistemima [1].

Ugrađeni računarski sistemi se odlikuju ograničenim i unaprijed definisanim skupom instrukcija za razliku od računarskih sistema opšte namjene koji omogućavaju platformu za praktično neograničen skup mogućih aplikacija. Praktična upotreba ugrađenih sistema je počela šezdesetih godina 20. vijeka primjenom u telefonskim centralama, uređajima i sistemima profesionalne (vojne) elektronike [1]. Prva upotreba u obliku sistema realizovanog sa integrisanim kolima je bila za upravljanje kosmičkim brodom *Apollo* [1]. Ugrađeni sistemi su sve više dolazili do izražaja sa smanjenjem fizičkih dimenzija integrisanih kola, povećanjem brzine i smanjenjem potrošnje. Revolucija je nastala sa pojavom mikrokontrolera koji su praktično ugrađeni sistemi na jednom integrisanom kolu. Ovo je značajno smanjilo fizičke dimenzije sistema, povećalo pouzdanost i brzinu rada, smanjilo cijenu i potrošnju. Osnovne karakteristike ugrađenih sistema su [1]:

- Većina ugrađenih sistema ima vremenska ograničenja za izvršavanje korisničkih programa;
- Često se koriste namjenske hardverske komponente;
- Hardverski resursi se unaprijed dimenzionišu prema poznatim softverskim potrebama;
- Velika pažnja se posvećuje upravljanju potrošnjom zato što se često radi o sistemima sa baterijskim ili solarnim napajanjem;
- Podjednaka pažnja se posvećuje i hardverskoj i softverskoj komponenti sistema.

Tokom perioda razvoja ugrađenih sistema razvijeno je mnogo softverskih alata za njihovo projektovanje i programiranje. Takođe, veliki je broj postojećih programskih

jezika prilagođen programiranju u ugrađenim računarskim sistemima.

Primjena operativnih sistema u ugrađenim računarskim sistemima donosi značajne prednosti kod projektovanja softvera i programiranja korisničkih programa: projektovanje softvera za konkretnu primjenu bez poznavanja detalja o hardveru sistema, visok stepen nezavisnosti programskog koda od konkretne hardverske platforme i postojanje mehanizama za istovremeno izvršavanje više korisničkih programa. U posljednje vrijeme se koristi grafički pristup projektovanju i programiranju ugrađenih sistema zasnovanih na operativnim sistemima, što omogućava projektovanje i programiranje bez poznavanja detalja o hardveru, operativnom sistemu i programskom jeziku.

U ovom radu prikazane su: osnovne karakteristike operativnog sistema RT Linuks, njegova upotreba u ugrađenim računarskim sistemima, opis praktične pripreme operativnog sistema za jednu hardversku platformu, kreiranje jednostavne korisničke aplikacije, pokretanje operativnog sistema na razvojnom okruženju, raspored zadataka koji se izvršavaju po prioritetima i mjerenje vremenskih parametara izvršavanja korisničke aplikacije pri različitim uslovima.

Operativni sistem je kreiran iz izvornog koda koji je javno dostupan. Za kompajliranje je korišten kompajler GCC, a korisnički program je napisan u programskom jeziku C. Korištena je razvojna ploča *Pandaboard ES* sa procesorom OMAP 4460 koji predstavlja sistem na integrisanom kolu (engl. *System on Chip-SoC*) [2].

II. OPERATIVNI SISTEM LINUX

Operativni sistem Linuks je razvijen 1991. godine u Helsinkiju od strane Linusa Torvaldsa. Razvijen je na osnovama operativnog sistema *Unix* poboljšavanjem programskog koda za samo jezgro operativnog sistema. Kod je zasnovan na programskom jeziku C. Izvorni kod je javno dostupan i njegova upotreba je moguća u skladu sa *GPL* i *BSD* licencama. Na osnovu istog jezgra razvijeno je više *desktop* operativnih sistema među kojima su najpoznatiji *Ubuntu*, *SUSE* i *Debian*.

Jezgro operativnog sistema Linuks je modularno i monolitno [1]. Modularnost se odlikuje time da se mogu koristiti samo komponente sistema koje su neophodne u konkretnoj primjeni. Ovo je posebno pogodno za ugrađene računarske sisteme, s obzirom da kod njih pretežno postoje ograničenja u pogledu memorije i prostora za čuvanje podataka i programa operativnog sistema. Monolitnost se odlikuje postojanjem odvojenih prostora za izvršavanje jezgra i izvršavanje korisničkih programa. Komunikacija između programa i jezgra se odvija posredstvom sistemskih poziva. Na ovaj način je onemogućen direktan pristup hardverskim resursima direktno iz aplikacije. Struktura jezgra prikazana je na slici 1.

Jezgro operativnog sistema Linuks se sastoji iz sljedećih blokova:

- sloj za apstrakciju hardvera (engl. *Hardware Abstraction Layer*);
- blok za upravljanje memorijskim resursima (engl. *Memory Management Module*);
- raspoređivač procesa i niti (engl. *Task and Thread Scheduler*);
- blok za upravljanje fajl-sistemima (engl. *File-System Manager*);
- blok sa drajverima uređaja (engl. *Device Driver Block*);
- mrežni podsistem (engl. *Networking Subsystem*);
- podsistem za međuprocenu komunikaciju (engl. *Inter-Process Communication Subsystem*).

Sloj za apstrakciju hardvera predstavlja interfejs između drajverskih funkcija i hardverskog sloja. Ovo omogućuje nezavisnost programskog koda drajvera od konkretne hardverske platforme.

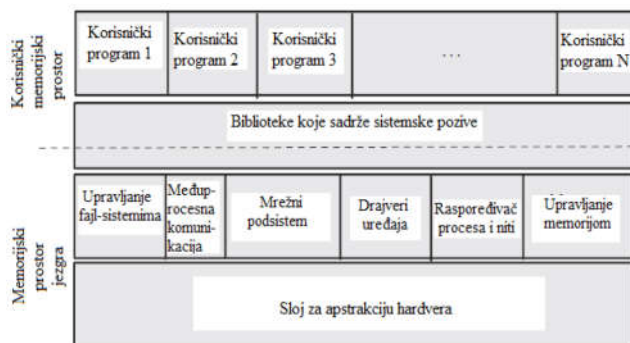
Blok za upravljanje memorijom upravlja pristupom memoriji i organizacijom memorije. Ovaj blok je odgovoran za dinamičko manipulisanje memorijom kod učitavanja i uklanjanja modula iz jezgra. Osim toga rezerviše memorijski prostor kod pokretanja korisničkih programa. Cjelokupan memorijski prostor je podijeljen na stranice veličine 4 kB.

Raspoređivač procesa i niti omogućava i upravlja višeprocensnim i višenitnim mogućnostima operativnog sistema. Procesi su pokrenuti korisnički program, a svaki ima jedinstveni identifikator na osnovu kojeg se prikazuje ostatku sistema i raspoređivaču procesa. Biblioteka programskog jezika C koja sadrži sistemske pozive za upravljanje raspoređivačem procesa.

Blok za upravljanje fajl-sistemima služi za upravljanje disk uređajima. Njegova funkcija je da virtueliše radnje kod pristupa disk uređaju. Na primjer, disk uređaj može biti tvrdi disk, optički disk ili memorijski stik, a operativnom sistemu se ovaj cjelokupni prostor predstavlja kao stablo sastavljeno od direktorijuma i fajlova.

Blok sa drajverima sadrži softver za upravljanje perifernim uređajima. Periferni uređaji se dijele na tri vrste: karakter uređaji, blok uređaji i mrežni uređaji. Karakter uređaji su oni sa kojima se komunikacija obavlja prenosom niza bita ili bajtova u jednom ili oba smjera. Primjeri ovih uređaja su *GPIO* priključci, serijski port i *SPI* komunikacioni moduli. Blok uređaji su oni koji služe za skladištenje fajl sistema. Mrežni uređaji su oni koji služe za komunikaciju po *IP* protokolu.

Svi periferni uređaji u operativnom sistemu Linuks se posmatraju kao fajlovi. Komunikacija korisničkog programa sa perifernim uređajem zasniva se na sistemskim pozivima za upis i čitanje podataka iz određenog fajla.

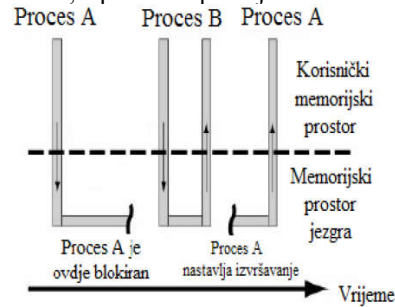


Slika 1. Blok šema memorijskog prostora korisničkih programa i jezgra [1]

Podsistem za međuprocenu komunikaciju upravlja komunikacijom između pokrenutih sistemskih i korisničkih aplikacija. Najčešće korišten fajl-sistem u operativnom sistemu Linuks je *ext3*.

III. OPERATIVNI SISTEM LINUXS ZA RAD U REALNOM VREMENU

Operativni sistem Linuks za rad u realnom vremenu ima mogućnost za pokretanje i izvršavanje programa u garantovanim vremenskim okvirima. Osobina programa ili procesa da bude blokiran od strane raspoređivača procesa i zamijenjen drugim procesom zove se prisvojivost. Proces sa višim prioritetom može zamijeniti proces sa nižim prioritetom prilikom izvršavanja. Slika 2 prikazuje izvršavanje i zamjenu izvršavanja procesa. Na ovoj slici je prikazano izvršavanje procesa A. U jednom trenutku proces B dobija viši prioritet, proces A se blokira, a proces B počinje sa izvršavanjem.



Slika 2. Primjer izvršavanja dva procesa tokom vremena [3]

Proces može imati tri stanja: može da bude u izvršavanju, može da bude blokiran od strane raspoređivača procesa i može da bude u stanju čekanja. Na osnovu vrste prioriteta, postoje tri vrste procesa: *SCHED_FIFO*, *SCHED_RR* i *SCHED_OTHER* [4]. Procesi *SCHED_FIFO* su procesi u realnom vremenu koji mogu imati prioritet od 1 (najniži) do 99 (najviši) i imaju viši prioritet od druge dvije vrste procesa. Jednom pokrenut proces ove vrste se izvršava sve dok ima viši prioritet od ostalih procesa ili dok dobrovoljno ne postane uspavan. Procesi *SCHED_RR* su procesi u realnom vremenu od 1 (najniži) do 99 (najviši). Oni su slični *SCHED_FIFO* procesima, s tim da svaki proces dobija određeni vremenski interval za svoje izvršavanje. Na primjer, ukoliko postoji pet ovih procesa sa prioritetom 15, četiri sa prioritetom 12 i tri sa prioritetom osam, prvih pet procesa će se izvršavati uzastopno u pet vremenskih intervala, poslije njihovog izvršavanja će se izvršavati četiri procesa sa prioritetom 12 u četiri uzastopna vremenska intervala, i na kraju tri procesa u tri uzastopna vremenska intervala. Procesi *SCHED_OTHER* su standardni procesi koji imaju podrazumijevani prioritet 0.

Važan parametar za procese u realnom vremenu je *jitter* koji označava najveću moguću varijaciju vremena u toku kojeg će se izvršiti pokrenuta stavka procesa. Upravljanje vremenskim karakteristikama procesa se obavlja sistemskim pozivima u programskom jeziku C. Najčešće korišteni sistemski pozivi su [6]:

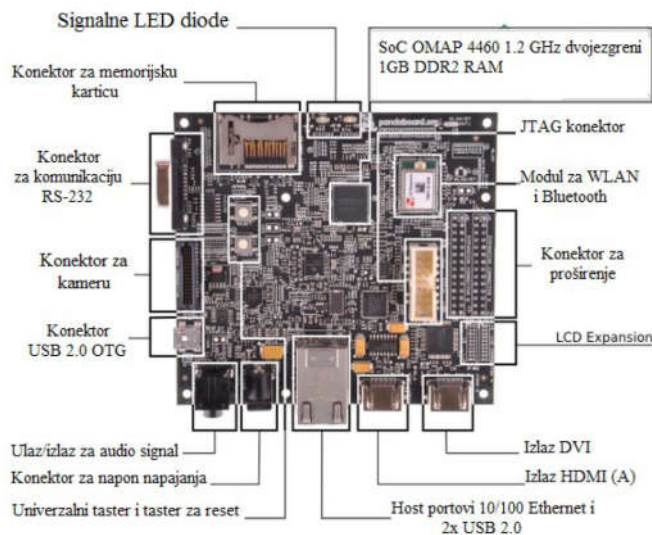
```

sched_setscheduler(PID,policy,*param)
sched_getscheduler(PID) i sched_yield()

```

IV. RAZVOJNO OKRUŽENJE PANDABOARD ES

Razvojna ploča *Pandaboard ES* sadrži procesor *OMAP 4460* koji ima dva jezgra, *DDR2 RAM* kapaciteta 1 GB, *Ethernet* port 10/100, modul za *Wi-Fi 802.11 b/g/n* i *Bluetooth*, komunikacioni interfejs *RS 232*, slot za memorijsku karticu, konektore za *HDMI*, *DVI*, *USB*, audio signale, kameru, konektor za *LCD* displej i interfejs koji je spojen sa *GPIO* priključcima. Ova ploča je prikazana na slici 3.



Slika 3. Razvojno okruženje Pandaboard ES

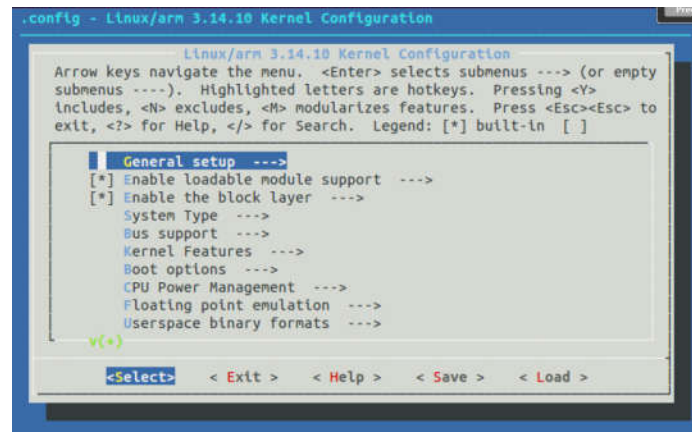
U procesu pokretanja sistema na ovom okruženju, postoje dvije faze. U prvoj se učitava *bootloader* prve faze, *X-Loader*. On učitava *bootloader* druge faze, *U-Boot*. *U-Boot* učitava jezgro. *Bootloaderi* i jezgro su smješteni na memorijskoj kartici. *U-Boot* učitava jezgro u radnu memoriju, podešava hardverske stavke i prosljeđuje dodatne argumente jezgru. Poslije toga, jezgro se spaja sa fajl-sistemom, nakon čega slijedi pokretanje sistemskih i korisničkih programa.

Na ovom razvojnom okruženju se koristi *SoC OMAP 4460* proizvođača *Texas Instruments*. Namijenjen je za multimedijalne primjene. Procesorska arhitektura je *ARM*, a postoji veše dodatnih procesora za specifične namjene u sklopu jednog integrisanog kola. Ovi procesori su prvenstveno bili namijenjeni za pametne telefone i tablete, ali je njihova namjena kasnije preorjentisana za ugrađene sisteme [2]. *OMAP 4460* je *OMAP* procesor četvrte generacije. Ima visoke multimedijalne karakteristike i zasnovan je na 45 nm tehnologiji. Omogućava obradu signala visoke definicije, obradu slika za dvodimenzionalne i trodimenzionalne video igrice i obradu slika veoma visoke rezolucije. *OMAP 4460* ima sljedeće glavne module: dvojezgreni *ARM* mikroprocesor *Cortex-A9*, procesor za digitalnu obradu signala, sistem za obradu slike i video signala visoke definicije, sistem za upravljanje displejima, dvojezgreni *ARM* mikroprocesor *Cortex-M3*, sistem za obradu zvučnog signala i sistem za obradu dvodimenzionalnog i trodimenzionalnog video signala. Procesor se nalazi u pakovanju koje ima 547 priključaka [5].

V. POKRETANJE OPERATIVNOG SISTEMA RT LINUXS NA RAZVOJNOM OKRUŽENJU

Jezgro i fajl sistem operativnog sistema Linuxs čuva se na memorijskoj kartici. *Bootloaderi* su takođe smješteni na istoj memorijskoj kartici. U praktičnom radu su korišteni računar sa operativnim sistemom *Ubuntu*, program *Gparted*, serijski kabal i kompajler *GCC*. Na operativnom sistemu *Ubuntu* obavljaju se sve pripremne operacije, program *Gparted* služi za pripremu memorijske kartice, a kompajler *GCC* se koristi za kompajliranje izvornog koda jezgra i korisničkih aplikacija. Na memorijskoj kartici su kreirane dvije particije, prva za bootloadere i jezgro operativnog sistema, i druga za fajl-sistem. Prva particija je formatirana u formatu *FAT32*, a druga u

formatu *ext3*. Fajlovi *MLO*, *u-boot.img* i *ulmage* snimaju se u prvu particiju, odakle se ti fajlovi očitavaju tokom procesa boot-ovanja. Na drugoj particiji je smješten fajl sistem. Za dibagovanje i kontrolisanje rada sistema koristi se komandna konzola preko serijskog interfejsa. Prilikom podešavanja parametara za kompajliranje izvornog koda za jezgro moguće je izabrati koji moduli će se koristiti. Grafički interfejs za ovo podešavanje je prikazan na slici 4.



Slika 4. Grafički interfejs za podešavanje modula jezgra [5]

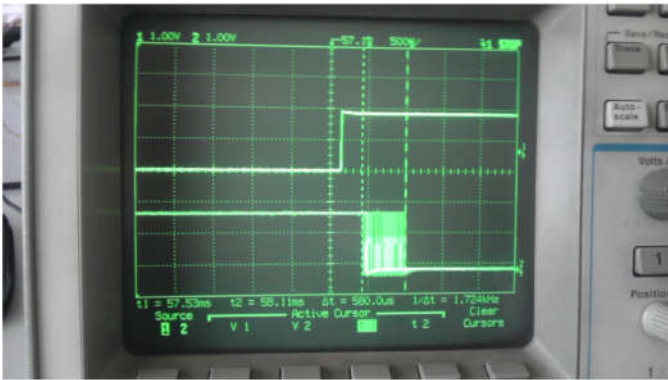
Korisnička aplikacija koristi jedan *GPIO* priključak za prikazivanje funkcionalnosti i sistemskih poziva za upravljanje ovim priključkom. *GPIO* priključak pripada vrsti karakter uređaja, što znači da se u operativnom sistemu predstavlja kao fajl. U suštini, kompletan pristup i podešavanje jednog *GPIO* priključka obavlja se preko više fajlova u okviru jednog direktorijuma za taj uređaj. Preko tih fajlova podešava se smjer signala i upisuje se vrijednost izlaznog signala ili se očitava ulazni signal. Pristup je moguć kroz komandnu konzolu ili iz korisničke aplikacije. Na primjer, dvije sljedeće konzolne komande podešavaju *GPIO* priključak kao izlazni i na tom izlazu se podešava visok logički nivo.

```
echo out > /sys/class/gpio/gpio110/direction
echo 1 > /sys/class/gpio/gpio110/value
```

Sljedeće dvije komande podešavaju isti *GPIO* priključak kao ulazni i sa njega očitava vrijednost signala.

```
echo in > /sys/class/gpio/gpio110/direction
cat > /sys/class/gpio/gpio110/value
```

Napisani korisnički program treba da prikaže osnovnu funkcionalnost dva *GPIO* priključka. Svaki pritiskom na taster S2 se mijenja stanje signala koji upravlja *LED* diodom D1. Električna šema ovih komponenti i odgovarajućih priključaka je prikazana na slici 5.



Slika 8. Prikaz vremenskih oblika ulaznog (gornji signal) i izlaznog (donji signal) u slučaju kada druga aplikacija ima viši prioritet [5]

Pokretanjem i izvršavanjem aplikacije u memorijskom postoru jezgra operativnog Sistema bi omogućilo smanjenje jittera i brže izvršavanje procesa. Ispitivanje vremenskih parametara prilikom izvršavanja aplikacije u memorijskom prostoru jezgra će biti tema narednog rada.

VII. ZAKLJUČAK

Projektovanje ugrađenog računarskog sistema upotrebom operativnih sistema značajno olakšava posao, jer je pažnja posvećena konkretnoj namjeni i programima, a ne direktnom podešavanju programskih registara, kao što je to slučaj kod projektovanja softvera isključivo u asemblerskom i nekim višim programskim jezicima. To znači da nije potrebno obimno proučavanje hardverske strukture konkretnog mikrokontrolera ili mikroprocesora.

Prednost ugrađenih računarskih sistema zasnovanih na operativnom sistemu RT Linux značajno olakšava realizaciju sistema koji zahtijevaju vremenske parametar za izvršavanje zadataka. Pravilnom prioritetizacijom aplikacija, kao i pravilnim podešavanjem dinamičke promjene i dodjele prioriteta, može se projektovati sistem koji ispunjava zahtjeve u pogledu vremenskih parametara. Međutim, za učenje teoretske

i praktične materije koja se odnosi na ovaj operativni sistem i njegovu upotrebu za ugrađene računarske sisteme potrebno je određeno vrijeme. Zaključak je da za relativno jednostavne namjene koje ne zahtijevaju snažne procesorske resurse treba koristiti mikrokontrolere i programiranje u višim programskim jezicima, dok za složene namjene, koje zahtijevaju snažne procesore i veliki broj radnji, treba koristiti sisteme na jednom integrisanom kolu i operativne sisteme u realnom vremenu.

LITERATURA

- [1] P. Raghavan, Amol Lad, Sriram Neelakandan, Embedded Linux System Design and Development, Auerbach Publications, Taylor & Francis Group, Boca Raton, New York, 2006. godina
- [2] OMAP 4460 Multimedia Device Technical Reference Manual, Texas Instruments, juli 2013. Godine
- [3] Christopher Hallinan, "Embedded Linux Primer: A Practical, Real-World Approach", Prentice Hall, septembar 2006. Godine
- [4] Karim Yaghmour, Jon Masters, Gilad Ben-Yossef, Philippe Gerum, Building Embedded Linux Systems, O'Reilly, avgust 2008. Godine
- [5] Boris Gluvić, Projektovanje ugrađenog računarskog sistema zasnovanog na operativnom sistemu Linuks, diplomski rad, Elektrotehnički fakultet u Banjoj Luci, 2014.
- [6] Daniel P. Bovet, Marco Cesati, "Understanding Linux Kernel", O'Reilly, novembar 2005. Godine

ABSTRACT

This paper presents operating system RT Linux deployment at embedded systems area. We described basic Linux modules and Linux OS characteristics. Linux real-time characteristics and Linux deployment on development environment Pandaboard ES are presented.

Keywords-Embedded systems; Microcontroller; Operating system;

TIME PARAMETERS CHARACTERISATION AT REAL-TIME LINUX BASED EMBEDDED SYSTEM

Boris Gluvić, Branko Dokić, Željko Ivanović