

Univerza v Ljubljani  
Fakulteta za računalništvo in informatiko

## **Operacijski sistemi 2**

### **Končno poročilo seminarske naloge MAIRM**

**Avtorja:**

Matej Jakop, 63070025

Gregor Kališnik, VNESI VPISNO

Ljubljana, 25. maj 2010

# Kazalo

<b>1</b>	<b>Opis izbrane teme in problemskega področja</b>	<b>2</b>
<b>2</b>	<b>Izbrani cilji, zahteve ter omejitve</b>	<b>2</b>
<b>3</b>	<b>Arhitektura rešitve</b>	<b>2</b>
<b>4</b>	<b>Izvedba rešitve in problemi pri tem</b>	<b>2</b>
<b>5</b>	<b>Morebitno nadaljnje delo</b>	<b>2</b>
<b>6</b>	<b>Navodila za uporabo</b>	<b>2</b>
6.1	Zahteve za delovanje sistema . . . . .	2
6.2	Potek namestitve . . . . .	3
6.2.1	Mobilni telefon . . . . .	3
6.2.2	Računalnik . . . . .	3
6.3	Povezava mobilnega telefona in računalnika ter pričetek delovanja . . . .	3
6.4	Uporabni napotki pri uporabi . . . . .	4
6.4.1	Miško premikamo ekranu v skladu s temi načeli: . . . . .	4
6.4.2	Novo kretnjo posnememo na sledeči način: . . . . .	4
6.4.3	Nastavitev kaj naj sistem naredi ob izvedbi določene kretnje . . .	4
6.4.4	Izboljšava zaznavanja kretenj . . . . .	5
	<b>Literatura</b>	<b>6</b>

# 1 Opis izbrane teme in problemskega področja

Tema seminarske naloge je izdelava virtualne naprave vhodno/izhodnega sistema, ki omogoča naslednje stvari:

- Upravljanje miške preko nagibanja telefona in pritiskanja tipk
- Vnašanje kratkih besedil preko tipkovnice mobilnega telefona
- Zaznavanje kretenj v 3D prostoru (nekaj narišemo s telefonom v zraku, sistem prepozna kaj to je in izvede ustrezno akcijo)

Če pogledamo malo bolj v podrobnosti in ne samo te točke ugotovimo, da izvedba celote ni tako trivialna kot se sprva zdi. Problematično je predvsem prepoznavanje kretenj v 3D prostoru.

## 2 Izbrani cilji, zahteve ter omejitve

## 3 Arhitektura rešitve

## 4 Izvedba rešitve in problemi pri tem

## 5 Morebitno nadaljnje delo

Projekt je objavljen na google code in kot tak je na voljo vsem, ki bi radi kaj dodali/spremenili. Torej odpte so vse opcije za sodelovanje drugih ljudi in tem uresničevanje njihovih idej. Kot uporabnika in razvijalca sistema sva prišla do ugotovitve, da bi bilo dobro še narediti sledeče stvari:

1. V XML datoteko uvesti pogoje glede na trenutno aktivno aplikacijo. Tako bi se lahko zagotovilo, da bi ista kretnja imela različen pomen v različnih programih. Npr. kretnja v desno bi v predvajalniku glasbe prestavila na drugo pesem, medtem ko v pregledovalniku slik na naslednjo sliko.
2. Nekako poskrbeti, da bi sistem sam izboljševal svoje znanje o kretnjah (brez sodelovanja uporabnika pri temu).
3. Prestaviti delovanje programa na višji prioritetni nivo (ker se drugače zna zgoditi, da ob veliki zasedenosti računalnika celoten sistem deluje upočasnjeno in ne tako kot prava miška).

Bistveno za nadaljnje delo je to, da morava poskrbeti, da bo več ljudi spoznalo najin projekt in ga morebiti začelo uporabljati. Le-tako bo smiselno dodajati nove funkcionalnosti in izboljšave. Če nimaš uporabnikov, vsak projekt slej ko prej umre.

## 6 Navodila za uporabo

### 6.1 Zahteve za delovanje sistema

Zahteve za delovanja sistema so skromne. Potrebno je imeti le:

- Operacijski sistem Windows/Linux/Mac
- računalnik z bluetooth vmesnikom
- mobilni telefon znamke Nokia s Symbian S60 3rd Edition ali Nokia s Symbian S60 5rd Edition, nameščeno python podporo (pyS60) in prisotnost merilnika pospeška v samem telefonu

Zadnja točka seznama se bo lahko s časom malce posplošila. V prihodnosti projekta bo verjetno dovolj, da bo uporaben poljuben telefon z merilnikom pospeška.

### 6.2 Potek namestitve

#### 6.2.1 Mobilni telefon

Možnih je več opcij:

1. V primeru že nameščenega pyS60 na mobilni telefon prenesemo (preko bluetooth ali kako drugače) samo mairm.py datoteko, ki se nahaja v paketu za inštalacijo. Na mobilnem telefonu jo shranimo na pomnilniško kartico v imenik data/python. Tako bo datoteka vidna v python vmesniku do interpreterja. Namestitev končana.
2. V primeru, da je v namestitvenem paketu prisotna datoteka mairm.sis le-to prenesemo na mobilni telefon in poženemo namestitev ter sledimo korakom na zaslonu.

#### 6.2.2 Računalnik

Na računalniku je namestitev sila preprosta. Potrebno je samo razpakirati namestitveni paket ali pognati namestitveno datoteko.

### 6.3 Povezava mobilnega telefona in računalnika ter pričetek delovanja

Za vzpostavitev povezave je potrebno slediti naslednjim korakom:

1. Vključimo bluetooth vmesnik v kolikor je izključen
2. Sparimo telefon in računalnik v primeru, da je eto potrebno storiti
3. Poženemo program MAIRM (če smo namestili z namestitvenim programom) oziroma v direktoriju namestitvenega paketa v konzoli poženemo ukaz

```
java -jar mairm.jar
```

4. Na mobilnem telefonu poženemo aplikacijo MAIRM (če smo namestili z mairm.sis) oziroma Python 2.x.x. Če smo pognali Python 2.x.x moramo v njem še odpreti skripto. To storimo tako, da izberemo optionsin nato run script. Poiščemo datoteko mairm.py in jo izberemo.
5. Prikaže se nam seznam bluetooth naprav in izberemo naš računalnik na katerem teče mairm. Po potrditvi tega nas bo telefon morebiti še povprašal po storitvi na katero se želimo povezati. Enostavno izberemo MAIRM.
6. Če so bili vsi koraki uspešno izvedeni, bi se sedaj morala miška začeti premikati po zaslonu v skladu s premikanjem telefona.

## **6.4 Uporabni napotki pri uporabi**

### **6.4.1 Miško premikamo ekranu v skladu s temi načeli:**

- Nagib telefona naprej, premakne miško navzgor
- Nagib telefona nazaj, premakne miško navzdol
- Nagib telefona levo, premakne miško levo
- Nagib telefona desno, premakne miško desno

TUKAJ PRIDE ŠE OSTALO, KAR ŠE NI NAREJENO ČISTO USKLAJENO ZA TIPKE NA TELEFONU

### **6.4.2 Novo kretnjo posnememo na sledeči način:**

1. Odpremo glavno okno aplikacije (če se aplikacija slučajno skriva v System tray področju)
2. Aplikacijo razširimo s klikom na sivi gumb, da dobimo opcijo vnosa imena kretnje
3. Vnesemo ime kretnje ter kliknemo na Learn
4. S pomočjo telefona (seveda mora biti najprej vzpostavljena povezava med mobilom in računalnikom) izvedemo kretnjo kot običajno
5. Če še nismo tega storili vsaj 5x se vrnemo 3. korak in ponovimo z istim imenom
6. Če še nimamo shranjeni vsaj 2 kretnji, ponovimo postopek vendar tokrat z drugo kretnjo.
7. Ko vse opravimo kot je bilo opisano v prejšnjih korakih, bi moral sistem začeti prepoznavati uporabnikove definirane kretnje

### 6.4.3 Nastavitev kaj naj sistem naredi ob izvedbi določene kretnje

1. Odpremo datoteko actions.xml, ki se nahaja v istem direktoriju, kot je bil nameščen sam program.
2. V datoteki bi že moral biti prisoten seznam vseh kretenj, ki jih sistem pozna.
3. Odločiti se je potrebno kaj bi radi, da sistem naredi:
  - izvede nek program  
V tem primeru med značke `<exec>` in `</exec>` vnesemo pot in ime programa.
  - pritisne določeno kombinacijo tipk  
V tem primeru med značke `<keys>` in `</keys>` dodamo nove tipke in sicer podobno kot v em primeru:

`<key delayRelease="false">F1</key>`

To pomeni, da se bo v primeru izvedbe kretnje simuliral pritisk tipke F1. Atribut `delayRelease` pa pomeni ali naj sistem drži tipko vse dokler se ne izvedejo še preostali pritiski tipk. Še en primer tega za pošiljanje windows+r (požene okno za zagon programa):

`<key delayRelease="true">WINDOWS</key>`

`<key delayRelease="false">r</key>`

Seznam vseh tipk, ki so na voljo za takšen vnos je zapisan v datoteki `keyboard_keys_available.txt` po prvem zagonu MAIRM programa.

4. Datoteko shranimo
5. Kliknemo z desno miškino tipko na ikonco mairm programa in izberemo opcijo Reload gesture action bindings:

### 6.4.4 Izboljšava zaznavanja kretenj

Če se opazi, da sistem dve kretnji pogosto zamenjuje potem lahko to rešite dokaj enostavno. Potrebno je posneti samo več ponovitev problematičnih kretenj (pri številu 15x ponovitev sistem že zelo dobro loči tudi skoraj čisto enake kretnje, npr. krog v levo in krog v desno).

# Literatura

- [1] Tasklets.[Online].[Citirano: 17. januar 2010; 16:00]. Dostopno na spletnem naslovu: <http://docs.blackfin.uclinux.org/doku.php?id=linux-kernel:tasklets>.
- [2] On using tasklets.[Online].[Citirano: 17. januar 2010; 16:22]. Dostopno na spletnem naslovu: <http://www.cs.usfca.edu/cruse/cs635/lesson17.ppt>
- [3] Deferred Procedure Call.[Online].[Citirano: 17. januar 2010; 17:00]. Dostopno na spletnem naslovu: [http://en.wikipedia.org/wiki/Deferred\\_Procedure\\_Call](http://en.wikipedia.org/wiki/Deferred_Procedure_Call)
- [4] Timer Objects and DPCs.[Online].[Citirano: 28. januar 2010; 16:00]. Dostopno na spletnem naslovu: <http://msdn.microsoft.com/en-us/library/aa490171.aspx>
- [5] Scheduling a recurring task in a device drive.[Online].[Citirano: 28. januar 2010; 17:00]. Dostopno na spletnem naslovu:[http://groups.google.si/group/mpc.lists.freebsd.hackers/browse\\_thread/thread/fffc2a918128&q=freebsd+tasklets#6a6edd1ae2654fc3](http://groups.google.si/group/mpc.lists.freebsd.hackers/browse_thread/thread/fffc2a918128&q=freebsd+tasklets#6a6edd1ae2654fc3)
- [6] Return-to-libc attack.[Online].[Citirano: 31. januar 2010; 13:00]. Dostopno na spletnem naslovu: [http://en.wikipedia.org/wiki/Return-to-libc\\_attack](http://en.wikipedia.org/wiki/Return-to-libc_attack)
- [7] Writing buffer overflow exploits - a tutorial for beginners.[Online].[Citirano: 27. januar 2010; 18:00]. Dostopno na spletnem naslovu: <http://mixter.void.ru/exploit.html>
- [8] Prosojnice s predavanj in zapiski
- [9] Avi Silberschatz,A;Galvin,P in Gagne,G.(2005).Operating System Concepts.