

Instytut Teleinformatyki

Wydział Inżynierii Elektrycznej i Komputerowej
Politechnika Krakowska



Laboratorium Administrowania Systemami Komputerowymi

„Środowisko testowe UML”

ćwiczenie numer: 1

Spis treści

1. WSTĘPNE INFORMACJE	3
1.1 TEMAT ĆWICZENIA	4
1.2 ZAGADNIENIA DO PRZYGOTOWANIA	4
1.3 CEL ĆWICZENIA	4
2. PRZEBIEG ĆWICZENIA	5
2.1 PRZYGOTOWANIE ĆWICZENIA	6
2.2 ZADANIE NR 1 – TWORZENIE I FORMATOWANIE DYSKÓW WIRTUALNYCH	7
2.3 ZADANIE NR 2 – INSTALACJA BAZOWYCH PAKIETÓW SYSTEMU WIRTUALNEGO	8
2.4 ZADANIE NR 3 – KONFIGURACJA ŚRODOWISKA SYSTEMU WIRTUALNEGO	9
2.5 ZADANIE NR 4 – URUCHOMIENIE WIRTUALNEGO ŚRODOWISKA	11
2.6 ZADANIE NR 5 – PORÓWNANIE WYDAJNOŚCI ŚRODOWISKA WIRTUALNEGO Z SYSTEMEM MACIERZYSTYM	12
2.7 ŚRODOWISKO GRAFICZNE SYSTEMU WIRTUALNEGO	14
2.8 ZAKOŃCZENIE ĆWICZENIA	15
2.9 OPRACOWANIE ĆWICZENIA I SPRAWOZDANIE	16

1. Wstępne informacje

1.1 TEMAT ĆWICZENIA

Tematem ćwiczenia jest tworzenie testowego środowiska systemowego.

1.2 ZAGADNIENIA DO PRZYGOTOWANIA

Przed przystąpieniem do wykonania ćwiczenia należy zapoznać się z następującymi zagadnieniami:

- Czym jest UML (User Mode Linux).
- Instalacja systemu Debian z wykorzystaniem debootstrap.

1.3 CEL ĆWICZENIA

Dzięki temu ćwiczeniu wykonujący pozna:

- sposób tworzenia testowego środowiska,
- instalację podstawowych pakietów systemu Debian,
- wydajność wirtualnego środowiska pracy.

2. Przebieg ćwiczenia

2.1 PRZYGOTOWANIE ĆWICZENIA

Uruchomienie komputera

Po załączeniu komputera należy uruchomić system operacyjny o nazwie ASK. Jest to dedykowany system umożliwiający wykonanie niniejszego ćwiczenia.

Logowanie

W celu wykonania ćwiczenia konieczne jest zalogowanie się na konto administratora (login: root, hasło: lab).

Katalog laboratoryjny

Przed przystąpieniem do zajęć należy utworzyć katalog gdzie będą przechowywane wszystkie pliki potrzebne do wykonania ćwiczenia.

```
stanowisko01:~/#mkdir uml
```

2.2 ZADANIE NR 1 – TWORZENIE I FORMATOWANIE DYSKÓW WIRTUALNYCH

Zadanie to polega na stworzeniu dwóch wirtualnych dysków, które będą później wykorzystywane przez system UML. Jeden z tych dysków zawierał będzie główny system plików, a drugi obszar wymiany.

W celu stworzenia pustych obrazów dysków wirtualnych należy wykorzystać narzędzie `dd`.

```
stanowisko01:~/uml# dd if=/dev/zero of=swapfs bs=1M  
count=128
```

```
stanowisko01:~/uml#dd if=/dev/zero of=rootfs.debian bs=1M  
count=1 seek=5000
```

Wykorzystanie takiej właśnie konstrukcji polecenia `dd` powoduje utworzenie pliku rzadkiego, tzn. takiego, który nie zajmuje miejsca w systemie plików. Proszę porównać dostępną przestrzeń dyskową za pomocą polecenia `df` przed i po utworzeniu pliku `rootfs.debian`. Proszę zauważyć, iż plik ten ma rozmiar 5GB.

Następnym krokiem będzie utworzenie systemów plików na wirtualnych dyskach:

```
stanowisko01:~/uml# mkswap swapfs
```

```
stanowisko01:~/uml# mkfs.ext4 rootfs.debian
```

2.3 ZADANIE NR 2 –INSTALACJA BAZOWYCH PAKIETÓW SYSTEMU WIRTUALNEGO

Teraz na przygotowanej partycji głównego systemu plików zainstalowane zostaną bazowe pakiety systemu Debian Etch.

Pierwszym krokiem jaki należy wykonać jest podmontowanie wirtualnego systemu plików, tak aby możliwe było zainstalowanie pakietów.

```
stanowisko01:~/uml# mkdir rootfs
```

```
stanowisko01:~/uml# mount -o loop rootfs.debian rootfs/ -t  
ext4
```

Kolejnym krokiem jest pobranie i zainstalowanie bazowych pakietów systemu Debian. Operacja ta trwa od kilku do kilkunastu minut w zależności od szybkości połączenia sieciowego. Czas ten proszę poświęcić na zapoznanie się ze stroną podręcznika polecenia debootstrap.

```
stanowisko01:~/uml# debootstrap --include=ssh,udev squeeze  
rootfs/ http://ftp.pl.debian.org/debian
```


2.4 ZADANIE NR 3 – KONFIGURACJA ŚRODOWISKA SYSTEMU WIRTUALNEGO

Pierwszą operacją jest konfiguracja interfejsów sieciowych systemu wirtualnego. W przypadku systemu debian konfiguracji interfejsów sieciowych dokonuje się poprzez edycje pliku `/etc/network/interfaces` znajdującego się w systemie wirtualnym.

```
stanowisko01:~/uml# vi rootfs/etc/network/interfaces
```

W pliku tym należy umieścić poniższe wpisy:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 192.168.0.2
    netmask 255.255.255.0
    broadcast 192.168.12.255
    gateway 192.168.0.1
```

Drugą operacją jest ustawienie nazwy wirtualnej maszyny. Nazwę maszyny definiuje się w pliku `/etc/hostname` znajdującym się w systemie wirtualnym.

```
stanowisko01:~/uml# vi rootfs/etc/hostname
```

W pliku tym należy umieścić poniższy wpis:

```
uml01
```

Trzecim krokiem konfiguracji środowiska wirtualnego jest określenie dostępnych systemów plików oraz ich parametrów montowania. Informacje te umieszcza się w pliku `/etc/fstab` znajdującym się w systemie wirtualnym.

```
stanowisko01:~/uml# vi rootfs/etc/fstab
```

W pliku tym należy umieścić poniższe wpisy:

```
/dev/ubda    /          ext4        defaults    0    1
/dev/ubdb    none       swap        sw          0    0
none         /proc     proc        defaults    0    0
none         /dev/shm  tmpfs       defaults    0    0
```

W czwartym kroku ustawione zostanie hasło administratora (użytkownik root). Hasło proszę ustawić na lab.

```
stanowisko01:~/uml# chroot rootfs passwd
```

W piątym kroku skopiowane zostaną moduły jądra systemu. Samo jądro oraz moduły instalowane są w systemie w momencie instalowania pakietu user-mode-linux.

```
stanowisko01:~/uml# cp -Rp /usr/lib/uml/modules/*  
rootfs/lib/modules/
```

Na koniec należy odmontować wirtualny system plików:

```
stanowisko01:~/uml# umount rootfs
```

Aby umożliwić komunikację sieciową wirtualnego środowiska, konieczne jest także wykonanie modyfikacji konfiguracji systemu macierzystego. Po pierwsze należy tak skonfigurować filtr pakietów iptables, aby maskował pakiety wysyłane z systemu wirtualnego na adres sieciowy systemu macierzystego. Po drugie należy włączyć przekazywanie pakietów.

```
stanowisko01:~/uml# iptables -t nat -A POSTROUTING -o eth0 -  
j MASQUERADE
```

```
stanowisko01:~/uml# echo 1 > /proc/sys/net/ipv4/ip_forward
```

2.5 ZADANIE NR 4 – URUCHOMIENIE WIRTUALNEGO ŚRODOWISKA

Po uruchomieniu wirtualnego środowiska otwarte zostaną wirtualne konsole, na których można logować się przy użyciu wcześniej ustawionego hasła.

```
stanowisko01:~/uml# linux root=/dev/ubda  
ubd0=./rootfs.debian ubd1=./swapfs eth0=tuntap,,192.168.0.1  
umid=uml01
```

Proszę zapoznać się ze stroną podręcznika polecenia linux.

2.6 ZADANIE NR 5 – PORÓWNANIE WYDAJNOŚCI ŚRODOWISKA WIRTUALNEGO Z SYSTEMEM MACIERZYSTYM

W tym zadaniu wykonane zostaną pewne testy mające na celu porównanie wydajności środowiska testowego w porównaniu z systemem macierzystym.

Test pierwszy polega na skopiowaniu 1TB danych z urządzenia /dev/zero do urządzenia /dev/null. Zadaniem urządzenia /dev/zero jest nieprzerwane generowanie ciągu zer. Z kolei urządzenie /dev/null jest „studnią bez dna”. Wszystko co zostanie tam przekierowane ginie bezpowrotnie.

Dodatkowo wykorzystane zostanie polecenie time, którego zadaniem jest wyświetlenie czasu jaki był potrzebny na wykonanie zleconej operacji. Proszę zapoznać się ze stroną podręcznika tego polecenia w celu zrozumienia zwracanych przez nie wartości.

Uwaga: aby testy były jak najbardziej miarodajne, konieczne jest wyłączenie systemu wirtualnego przed wykonywaniem testów systemu macierzystego (polecenie shutdown -h now). Aby uniknąć także błędów przypadkowych proszę testy powtórzyć kilkakrotnie.

Dla systemu macierzystego:

```
stanowisko01:~/uml# time dd if=/dev/zero of=/dev/null bs=1M  
count=1000000
```

Dla systemu wirtualnego:

```
uml01:~# time dd if=/dev/zero of=/dev/null bs=1M  
count=1000000
```

Drugi test polega na utworzeniu pliku o wielkości 128M. Do tego celu ponownie wykorzystane zostanie polecenie time, dd oraz urządzenie /dev/zero.

Dla systemu macierzystego:

```
stanowisko01:~/uml# time dd if=/dev/zero of=./test bs=1M  
count=128
```

Dla systemu wirtualnego:

```
uml01:~# time dd if=/dev/zero of=./test bs=1M count=128
```

Trzecim testem jest test mocy obliczeniowej. Chodzi o sprawdzenie wydajności procesora w systemie wirtualnym. Proszę napisać prosty, lecz czasochłonny program w języku C (najlepiej program ze skomplikowaną pętlą oraz dużą ilością obliczeń matematycznych), a

następnie skompilować go oraz uruchomić w obu systemach. Czas wykonywania takiego programu powinien być dłuższy niż 1 minuta.

Aby możliwa była kompilacja programu w systemie wirtualnym konieczne będzie zainstalowanie dodatkowych pakietów. W przypadku systemu wirtualnego operację tą wykonuje się w taki sam sposób jak w przypadku macierzystego systemu.

```
uml01:~# apt-get install gcc libc6-dev
```

Podczas testowania wydajności obliczeniowej proszę do pomiaru czasu wykonywania stworzonego programu używać polecenia `time`.

2.7 ŚRODOWISKO GRAFICZNE SYSTEMU WIRTUALNEGO

Aby umożliwić uruchamianie aplikacji graficznych w systemie wirtualnym konieczne jest wykorzystanie X serwera systemu macierzystego. Poniższe polecenie powoduje uruchomienie drugiej instancji serwera.

```
stanowisko01:~/uml# X -ac :1
```

Przełączanie pomiędzy 2 instancjami serwera X odbywa się z wykorzystaniem kombinacji klawiszy Ctrl+Alt+F7, Ctrl+Alt+F8.

W systemie wirtualnym konieczne jest zainstalowanie menadżera X Window.

```
uml01:~# apt-get install icewm xterm
```

Następnie konieczne jest odpowiednie ustawienie zmiennej DISPLAY (adres IP uzależniony jest od numeru stanowiska, ostatni oktet jest taki jak numer stanowiska).

```
uml01:~# export DISPLAY="192.168.112.<nr>:1"
```

Uruchomienie menadżera X Window.

```
uml01:~# icewm
```

Po przełączeniu się na drugą instancję X serwera (Ctrl+Alt+F8) dostępny jest graficzny pulpit wirtualnego systemu.

2.8 ZAKOŃCZENIE ĆWICZENIA

Po zakończeniu ćwiczenia proszę usunąć wszystkie pliki i katalogi które zostały utworzone podczas wykonywania ćwiczeń.

Jeżeli ćwiczenie wykonywane było zgodnie z instrukcją, na zakończenie ćwiczenia konieczne jest wyłączenie systemu wirtualnego

```
uml01:~# shutdown -h now
```

oraz usunięcie w systemie macierzystym katalogu uml.

```
stanowisko01:~/uml# cd
```

```
stanowisko01:~# rm -rf ~/uml/
```

2.9 OPRACOWANIE ĆWICZENIA I SPRAWOZDANIE

Wykonanie ćwiczenia polega na praktycznej realizacji wszystkich zadań **Rozdziału 2** niniejszej instrukcji zatytułowanego „**Przebieg Ćwiczenia**”. Należy sporządzić sprawozdanie z wykonania ćwiczenia (w formie dokumentu elektronicznego) i w ciągu najdalej dwóch tygodni od dnia wykonania ćwiczenia oddać je prowadzącemu zajęcia.

Kompletne opracowanie ćwiczenia powinno zawierać:

- ✓ Część opisową odnoszącą się do teorii przerabianego ćwiczenia. Ta część sprawozdania powinna wykazać dobrą ogólną znajomość zagadnień leżących u podstaw przerabianego tematu, znajomość odnośnej literatury, samodzielność myślenia i umiejętność pisania opracowań o charakterze technicznym.
- ✓ Wnioski praktyczne wynikające z wykonania ćwiczenia, a w tym:
 - uwagi odnoszące się do przebiegu ćwiczenia (np. czy dane ćwiczenie może być wykonane z pełnym rozumieniem zawartych w nim czynności i problemów, czy ćwiczenie jest możliwe do wykonania w czasie przeznaczonym na zajęcia, czy ćwiczenie jest zbyt trudne/ zbyt łatwe, itp.,
 - uwagi odnoszące się do sposobu przygotowania i jakości (waloru dydaktycznego) instrukcji do ćwiczenia,
 - uwagi odnoszące się do ewentualnych utrudnień technicznych lub organizacyjnych pojawiających się w trakcie wykonywania ćwiczenia,
 - postulaty merytoryczne i techniczne dotyczące usprawnienia/ulepszenia jakości wykonywanego ćwiczenia,
 - inne

Wnioski z drugiej części sprawozdania posłużą do usprawnienia i poprawy zajęć laboratoryjnych w latach następnych.