



Systemy Operacyjne

Automatyzacja skryptowa

Dr hab. inż. Krzysztof Rzecki, prof. AGH



Sprawy organizacyjne

- Prowadzący: **Krzysztof Rzecki**, <http://rzecki.pl>
- Wykłady: 30h
- Laboratorium: 15h
- ECTS: 3.0
- Warunki zaliczenia:
 - Warunkiem zaliczenia wykładu jest pozytywna ocena z kolokwium teoretycznego.
 - Warunkiem zaliczenia laboratorium są pozytywne oceny z odpowiedzi oraz kolokwium praktycznego.
 - Warunkiem zaliczenia przedmiotu są trzy pozytywne, w/w oceny.
- Sposób obliczania oceny końcowej
 - Ocena wystawiana na podstawie średniej ważonej wyników uzyskanych z:
 - kolokwium teoretycznego z zakresu wykładów (30%),
 - kolokwium praktycznego z zakresu ćwiczeń laboratoryjnych (40%),
 - ocen[-a]-y z odpowiedzi ustnej uzyskanej w trakcie ćwiczeń laboratoryjnych (30%).

BIO

Edukacja / Nauka

- **Informatyka:**
 - inżynier i magister - EAIiE AGH,
 - doktor - IITiS PAN,
 - habilitacja - RD ITiT AGH.
- **Zarządzanie i Marketing:**
 - magister - WZ AGH.

Certyfikaty: POWR.3.5, TOP 500 Innovators, ITIL Foundation, PRINCE2.

Badania

- metody biometryczne, gesty,
- systemy wizyjne i bronchoskopia,
- radioterapia i promieniowanie kosmiczne,
- laser-induced breakdown spectroscopy,
- wirtualizacja,
- świadomość kontekstu,
- architektury zorientowane na usługi,
- protokoły sieciowe.

Zawód / Przemysł / Biznes

- Profesor uczelni - KBiB, EAIIB, AGH.
- CEO - Live-Docs.com Sp. z o.o., Kraków.
- Kierownik projektu - KI, IMF, PK, Kraków.
- Kierownik B+R - EPL Sp. z o.o., Międzyrzecz.
- Recenzent w NCBiR.

Doświadczenie

- ComArch Healthcare S.A., Kraków,
- VSoft SA, Kraków,
- CCNS SA, Kraków,
- Telekomunikacja Polska SA, Warszawa,
- Siemens AG, Monachium,
- Nokia Siemens Networks AG, Monachium.

Programy: Fulbright Cybersecurity, ISS on DL.

Linked in:

<https://www.linkedin.com/in/krzysztof-rzecki/>



Systemy operacyjne

1. Wprowadzenie i podstawy automatyzacji skryptowej.
2. Rodzaje i architektura systemów operacyjnych.
3. Program, proces, wątek, tworzenie i terminowanie, stany, współbieżność i równoległość.
4. Komunikacja międzyprocesowa, synchronizacja, zakleszczenia.
5. Pamięć operacyjna, pamięć główna, pamięć wirtualna, przestrzeń wymiany, zarządzanie pamięcią.
6. Magazyn danych i system plików.
7. Gniazda i komunikacja sieciowa.
8. Ochrona i bezpieczeństwo systemów operacyjnych.

Co w ramach tego przedmiotu byłoby interesujące, a nie mieści się na w/w liście?



Literatura

- A. Silberschatz, P. B. Galvin, G. Gagne, Operating System Concepts Essentials, 10th ed.
- A. S. Tanenbaum, Bos H., Modern Operating Systems, Pearson, 4th ed.
- Cooper M., Advanced Bash-Scripting Guide. An in-depth exploration of the art of shell scripting, 2014, online: <https://www.tldp.org/LDP/abs/html/>
- W. Richard Stevens, Stephen A. Rago, Advanced Programming in the UNIX Environment, 3rd ed.
- W. R. Stevens, UNIX Network Programming, 2nd ed.
- R. Love, Linux System Programming: Talking Directly to the Kernel and C Library, 1st ed.
- P. Yosifovich, M. E. Russinovich, D. A. Solomon, A. Ionescu, Windows Internals, Part 1: System architecture, processes, threads, memory management, and more, 7th ed.



Przygotowanie środowiska

- VPN do sieci AGH
 - <https://pomoc-it.agh.edu.pl/vpn-zdalny-dostep-do-sieci/>
- Konto na serwerze studenckim
 - <https://pomoc-it.agh.edu.pl/konta-unix/instrukcje/zakladanie-konta-unix/>
- Własna instalacja systemu Linux (2+ cores of CPU, 4+ GB RAM, 25+ GB HDD)
 - Natywna, czyli bezpośrednio na komputerze
 - Maszyna wirtualna:
 - VirtualBox: <https://www.virtualbox.org/>
 - QEMU: <https://www.qemu.org/>
 - VMware Player: <https://www.vmware.com/products/workstation-player.html>
- **Uwaga!** Rozwiązania w postaci terminala MacOS, PowerShell, czy Windows Subsystem for Linux(WSL) są niewystarczające do realizacji ćwiczeń.



Przygotowanie środowiska do zajęć on-line

- **Sprawdź działanie swojego systemu audio** - połącz się przed zajęciami z Koleżanką/Kolegą za pomocą aplikacji, którą użyjesz do zajęć on-line i zapytaj, czy Cię słyszy, wyreguluj ustawienia mikrofonu tak, aby w czasie zajęć było Cię odpowiednio słychać, ale nie przesteruj wzmocnienia.
- **Sprawdź działanie kamery** - połącz się j.w. i zapytaj, czy Cię widać.
- **Sprawdź działanie udostępniania pulpitu i okna aplikacji** - połącz się j.w. i przetestuj działanie udostępniania całego pulpitu oraz poszczególnych okien.

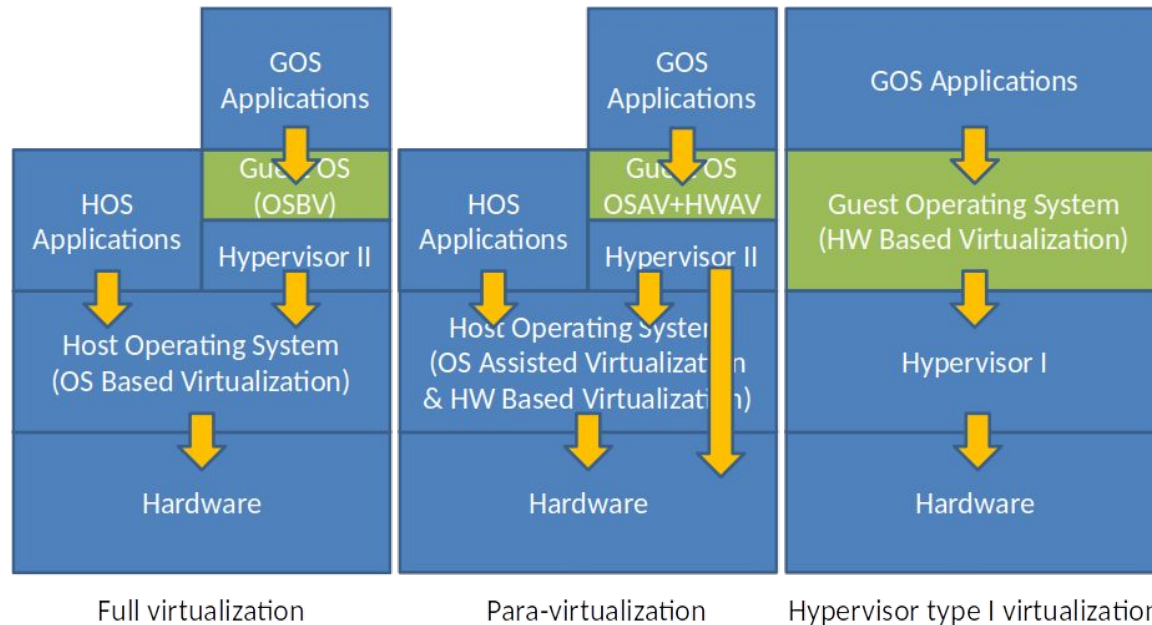


Własna instalacja systemu Linux

Najpopularniejsze dystrybucje:

- Debian
- Ubuntu
- **Mint Cinnamon**
- Red Hat / CentOS
- Fedora
- openSUSE
- Mandriva
- Slackware

Wirtualizacja





Implementacje wirtualizacji

Pełna wirtualizacja Hipernadzorca II typu	Para-wirtualizacja	Bare Metal Hypervisor Hipernadzorca I typu
<ul style="list-style-type: none">• Wirtualizacja sprzętowa BT (Binary Translation):<ul style="list-style-type: none">◦ VirtualBox, 32bit host,◦ VMware Workstation, 32bit,◦ VMware Server,◦ Parallels Desktop.• Wirtualizacja sprzętowa VT (Virtualization Technology): VT-x, AMD-V<ul style="list-style-type: none">◦ VMware Workstation, 64bit,◦ VirtualBox, 64bit host,◦ QEMU.	<ul style="list-style-type: none">• VirtualBox (tryb parawirtualizacji)• Xen (tryb parawirtualizacji)• Oracle VM (baza: Xen)• IBM LPAR	<ul style="list-style-type: none">• KVM (moduł jądra Linux)• Microsoft Hyper-V• VMware ESXi• Xen• Oracle VM Server (baza: Xen) <p>Również implementacje wspierane sprzętowo VT, z pierwszej kolumny:</p> <ul style="list-style-type: none">◦ VMware Workstation, 64bit,◦ VirtualBox, 64bit host.



Wirtualizacja

- Pełna wirtualizacja – binarna translacja instrukcji z systemu gościa do sprzętu poprzez system hosta:
 - Wirtualizowany system operacyjny nie wymaga żadnych zmian/modyfikacji.
 - Większe bezpieczeństwo w izolacji awarii.
- Parawirtualizacja – część instrukcji i odwołań systemu gościa jest tłumaczonych, a część jest przekazywana natywnie do sprzętu:
 - Efektywniejsze wykorzystanie współdzielonych zasobów.
 - Mniejszy narzut obliczeniowy na hosta.



Podstawowe polecenia - przypomnienie

`pwd, cd, ls, cat, cp, mv, mkdir, rm, touch, locate,`

`find, grep, df, du, head, tail, diff, tar, chmod, chown,`

`id, jobs, kill, ping, wget, history, man, echo, zip, unzip,`

`hostname, useradd, userdel, curl, df, diff, echo, exit, finger, free,`

`grep, groups, less, passwd, ping, shutdown, ssh, reboot, sudo, top,`

`uname, w, whoami`



Hello world...

```
#!/bin/bash
```

```
echo "Hello world..."
```

Advanced Bash-Scripting Guide:

<https://tldp.org/LDP/abs/html/>

Uprawnienia do plików

```
krz@zinc:~/abc$ ls -al
razem 44
drwxrwxr-x  2 krz krz  4096 paź  9 18:40 .
drwx----- 98 krz krz 28672 paź  9 18:40 ..
-rw-rw-r--  1 krz krz    0 paź  9 18:40 file.txt
krz@zinc:~/abc$
```

- Katalog bieżący: ~/abc oraz .
- Katalog nadrzędny: ..

Ustawianie uprawnień:

```
$ chmod uprawnienia plik
$ chmod 644 file.txt
$ chmod a+rx,a-w directory
```

Uprawnienia, przykład:

```
drwxr-x---
0123456789
```

Pozycja 0: d (dir), l (link), b (block), c (character)

Pozycja 1, 2 i 3: uprawnienia właściciela 'u'

Pozycja 4, 5 i 6: uprawnienia grupy 'g'

Pozycja 7, 8 i 9: uprawnienia pozostałych 'o'

Pozycje 1..9: uprawnienia wszystkich 'a'

rwX - read, write, eXecute

421 - zapis numeryczny,

np. r-x = 5, rw- = 6, r-- = 4.

Uprawnienia do plików

0	---	brak uprawnień	blokada
1	--x	wykonywanie	nieprzydatne
2	-w-	zapis	zbieranie sekretnych logów
3	-wx	zapis i wykonywanie	nieprzydatne
4	r--	odczyt	stała konfiguracja
5	r-x	odczyt i uruchamianie	pliki wykonywalne
6	rw-	odczyt i zapis	pliki edytowalne
7	rwX	odczyt, zapis i uruchamianie	skrypty i programy
	??s	bit suid	programy specjalne

Uprawnienia do katalogów

0	---	brak uprawnień	blokada
1	--x	wykonywanie	katalog bez podglądu
2	-w-	zapis	nieprzydatne
3	-wx	zapis i wykonywanie	nieprzydatne
4	r--	odczyt	nieprzydatne
5	r-x	odczyt i uruchamianie	katalogi bez możliwości zmian zaw.
6	rw-	odczyt i zapis	nieprzydatne
7	rwX	odczyt, zapis i uruchamianie	katalogi z możliwością zmian zaw.
	??t	sticky bit	katalog specjalny



Uprawnienia do uruchomienia

Podstawowe uprawnienie do wczytania i uruchomienia skryptu:

```
$ chmod 555 skrypt
```

lub:

```
$ chmod +rx skrypt
```

Sprawdź:

```
$ chmod u+s skrypt
```



Skrypty - podstawy

Przypisanie wartości do zmiennej:

```
$ a=5  
$ b=$a
```

Przypisanie wyniku działania polecenia do zmiennej:

```
$ a=`ls /var/log | wc -l`    lub nowsze:    a=$(ls /var/log | wc -l)
```

Wypisanie wartości zmiennej na ekran:

```
$ echo "Wartość a wynosi $a"
```



Wybrane znaki specjalne

* wildcard

```
$ ls /usr/*bin/
```

? test operator

```
$ variable = a<10?5:7
```

{a,b,c} rozwijanie

```
$ file.{bin,txt} == file.bin file.txt
```

```
{a..z} lub {1..9}
```

\$[...] obliczenie

```
$ a=5; c=${a+5} lub: $ let c=${a+5}
```

\$ oznaczenie zmiennej

```
$ a=5; echo $a
```



Operacje na zmiennych

Podstawienie

```
$ a='Ala ma kota'  
$ b=${a/Ala/Piotr}  
$ echo $b
```

Wyzerowanie zmiennej:

```
$ a=' '      lub  a=      lub  a=""
```

Ćwiczenie:

```
$ a=5  
$ a+=6      vs.  $ let a+=6  
$ echo $a
```



Warunek if

```
if [ condition ]  
then  
    command  
    ...  
fi
```

```
if [ condition ]  
then  
    command  
    ...  
elif [ condition ]  
    command  
    ...  
else  
    command  
    ...  
fi
```

```
if [ condA ] && [ condB ]  
then  
    command  
    ...  
elif [ condC ] || [ condD ]  
    command  
    ...  
else  
    command  
    ...  
fi
```



Petla for

```
for arg [list]
do
    command
...
done
```

```
for n in one two
do
    command
...
done
```

```
for n in {1..5}
do
    command
...
done
```

```
for n in "a b" "c d"
do
    Set -- $n
    cmd with $1
    cmd with $2
...
done
```




Pęta while

```
while [ condition ]  
do  
    command  
    ...  
done
```

```
while [ $a -lt $b ]  
do  
    command  
    ...  
done  
  
lt, le, gt, ge
```

```
cond()  
{ if ... return 0;  
  else return 1;}  
  
while cond  
do  
    command  
    ...  
done
```



Pętla until

Czy jest sterowana odwrotnym warunkiem niż while ?



Podnoszenie uprawnień: su, sudo (doas)

Zalogowany jako 'root':

```
# id          -> uid=0(root) gid=0(root) grupy=0(root)
```

Zalogowany jako użytkownik:

```
$ id          -> uid=1000(krz) gid=1000(krz) grupy=1000(krz),...
```

\$ su zmiana id użytkownika na 0 (superużytkownik, root)
wymagane: hasło użytkownika root
wykonanie pojedynczego polecenia: \$ su -c <polecenie>

\$ sudo wykonanie polecenia z uprawnieniami root'a
wymagane: hasło bieżącego użytkownika + uprawnienia w /etc/sudoers
uzyskanie powłoki: \$ sudo -i



Podnoszenie uprawnień: su, sudo - c.d.

```
$ sudo id          -> uid=0(root) gid=0(root) grupy=0(root)
$ su - -c "id"-> uid=0(root) gid=0(root) grupy=0(root)
```

Aby możliwe było użytkowanie polecenia `su` należy ustawić hasło dla `root`:

- Typowa instalacja Linux: hasło ustawione jest podczas instalacji systemu.
- Instalacja Ubuntu i pochodnych: hasło trzeba ustawić poprzez `sudo`:

```
$ sudo passwd root
```

Aby możliwe było korzystanie z polecenia `sudo` należy ustawić uprawnienia (`visudo`):

- Typowa instalacja Linux: jest konto 'root', brak ustawień `sudoers`.
- Instalacja Ubuntu i pochodnych: użytkownik konfigurowany w trakcie instalacji ma uprawnienia.



Podnoszenie uprawnień: su, sudo - c.d. 2

Połączenie poleceń:

```
$ sudo su          - mając w /etc/sudoers uprawnienia do sudo bez hasła przejdziemy do 'root'  
$ sudo -i
```

Taki sposób **nie wywołuje** wykonania `.bashrc` konta root:

```
$ su -c <polecenie>  
$ sudo polecenie
```

Taki sposób **wywołuje** wykonanie `.bashrc` konta root (odpowiednik: `$ source .bashrc`):

```
$ su - -c <polecenie>  
$ sudo su - -c <polecenie> !!!
```



Czerwone Wierchy