

1

Introducere

26 februarie 2009

- Prezentarea generală a cursului
- The kernel world
- Arhitectura Linux
- Arhitectura Windows NT

- Octavian Purdilă, Răzvan Deaconescu, Dragoș Tătulea, George Milescu
- Mult succes în noul semestru!

application programming (EGC, SPG, PP, SPRC, IOC, etc.)

system programming (PC, SO, PT)

user
space

kernel programming (PSO)

kernel
space

- Site: <http://cs.pub.ro/~pso>
 - Wiki
- 10 puncte teme de casă
 - Linux, Windows, Linux si Windows
 - Punctajul > 5 puncte e corelact direct proportional cu nota de la examen
 - Tema copiată -> punctaj 0 la toate temele
- 2 puncte activitate la laborator
 - nu se punctează doar prezența
- 3 puncte „examen”
 - Teste la curs (~9)
 - 3 întrebări, open book
 - Testează înțelegerea conceptelor
- Condiții de promovare: nota finala 4.5, nota minima examen 3

- Prezentarea structurii interne a unui sistem de operare
- Target: sisteme de operare de uz general
- Structura și componentele unui kernel monolitic
 - Procese, FS, Networking
 - Memory management
- Exemplificare pe Linux și NT

- Însușirea cunoștințelor necesare implementării de device drivere
- Înțelegerea în profunzime a cunoștințelor prin rezolvarea de exerciții

- Programare: C
- SDA: tabele de dispersie, arbori balansați
- PLAS: lucrul cu regiștri și instrucțiuni de bază (adunări, comparații, salturi)
- CN: TLB/CAM, cache
- PC, RC: ethernet, IP, sockete
- SO: procese, fișiere, thread-uri, memorie virtuală

- ~ 12 cursuri
- interactiv
 - participați la discuții
 - întrebați atunci când nu ați înțeles
- destul de “dens”, se recomandă călduros parcurgerea suportului bibliografic înainte și după curs

- Introducere
- Procese
- Scheduling
- Apeluri de sistem
- Traps
- Spațiul de adresă
- Memorie virtuală
- Memorie fizică
- Block I/O
- Sisteme de fișiere (1)
- Sisteme de fișiere (2)
- SMP
- Securitate
- Networking

- Device Drivers (C)
- 20 min prezentare / 80 minute lucru
- se punctează activitatea
- learn by doing
- maxim 15 puncte pe laborator
- bonusul poate compensa activitatea slabă la alte laboratoare și cel mult un laborator pierdut (absență)

- Monitorizare apeluri de sistem
- Driver pentru portul serial
- Software RAID
- Driver pentru un sistem de fișiere simplu
- Statefull firewall

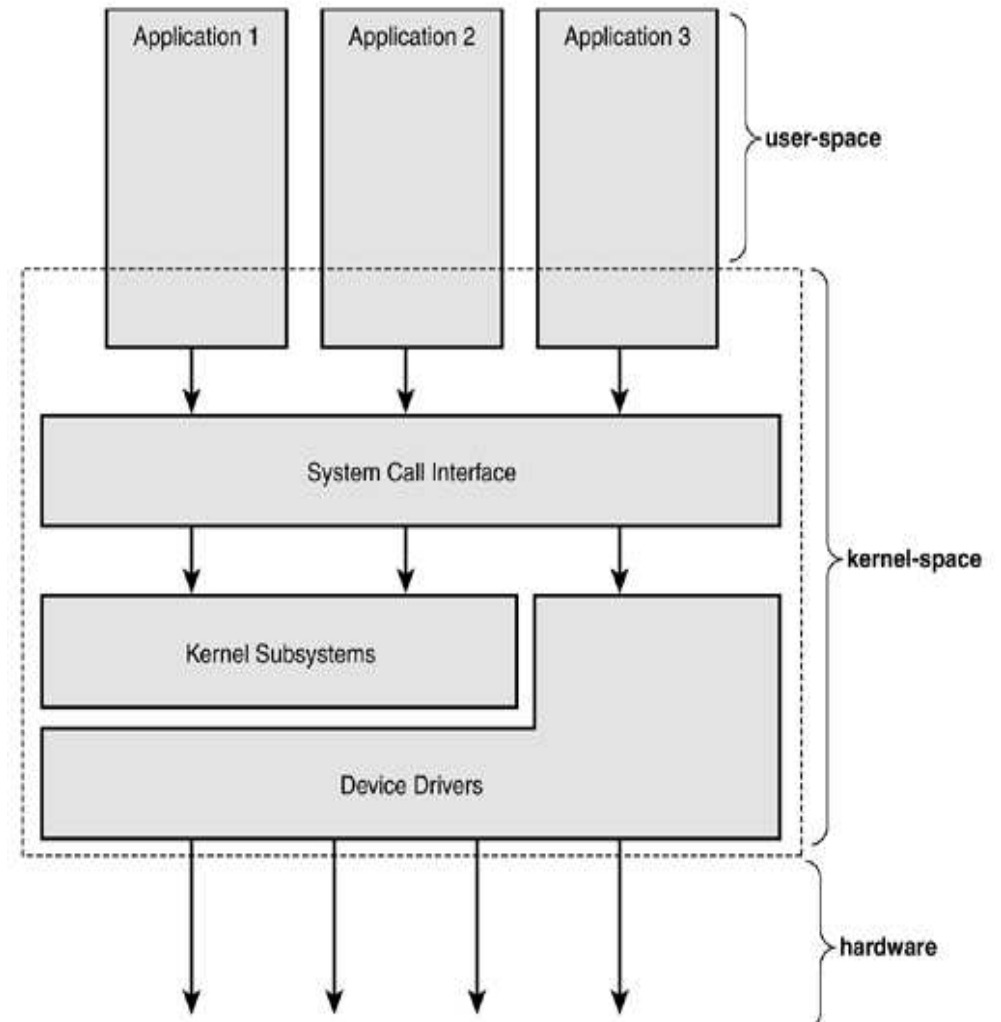
- necesare: aprofundare API (laborator) și concepte (curs)
- teste publice
- suport de testare
- relativ puțin cod de scris dar relativ dificile
- dificultatea constă în acomodarea cu noul mediu
- sunt deja afișate pe site! :-)

- Linux Kernel Development, 2nd edition, Robert Love, Noval Press 2005
- Windows Internals 4th edition, David A. Solomon & Mark E. Russinovich, Microsoft Press 2005
- Understanding the Linux Kernel, 3rd edition, Daniel P. Bovet & Marco Cesati, O'Reilly 2005
- Linux Networking Architecture, Klaus Wehrle, Frank Pahlke, Hartmut Ritter, Daniel Muller, Marc Bechler, Prentice Hall 2004
- Understanding Linux Network Internals, Christian Benvenuti, O'Reilly 2005

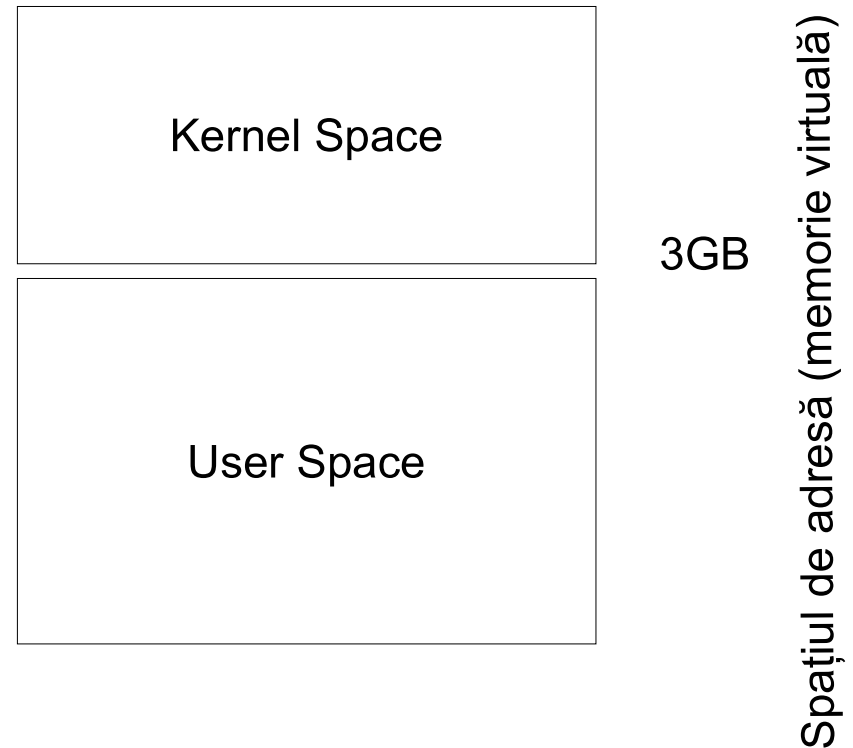
- Linux Device Drivers, 3nd edition, Alessandro Rubini & Jonathan Corbet, O'Reilly 2006
- Programming the Windows Driver Model, 2nd edition, Walter Oney, Microsoft Press, 2002
- The Windows 2000 Device Driver Book, 2nd edition, Art Baker & Jerry Lozano, Prentice Hall, 2001
- Linux Kernel in a Nutshell, Greg Kroah-Hartman, O'Reilly 2005

- UTLK: capitolul 1
- LKD: capitolul 1
- WI: capitolele 1 și 2

- Aplicații
- Sistem de operare
 - Shell, utilitare de bază
 - Kernel

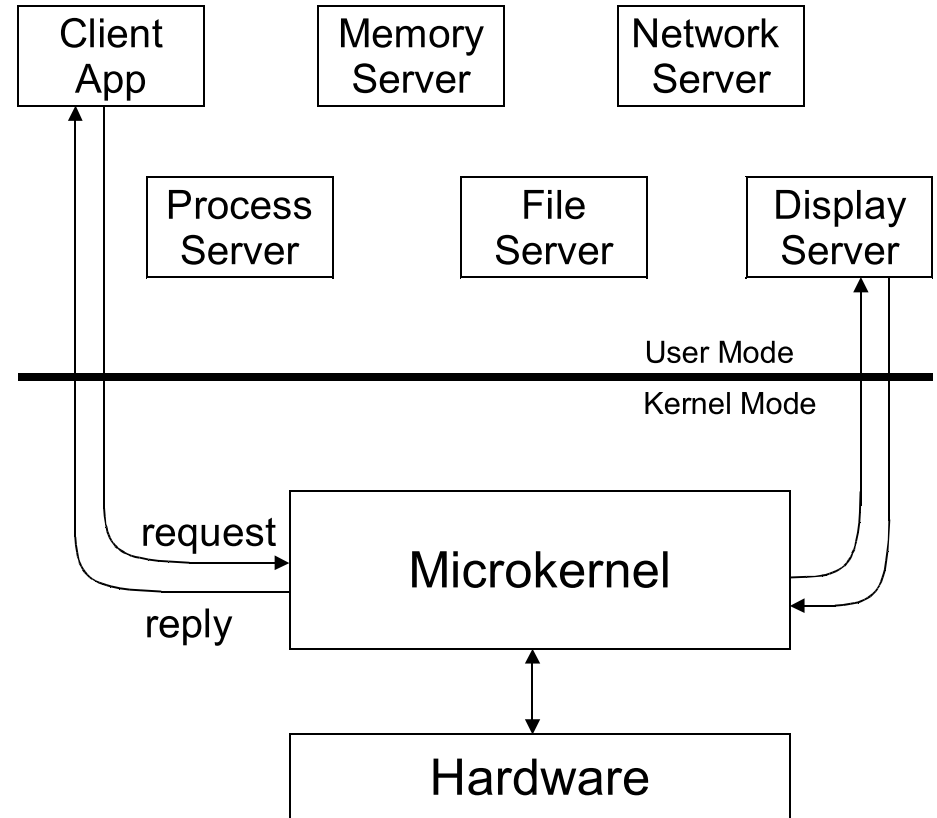
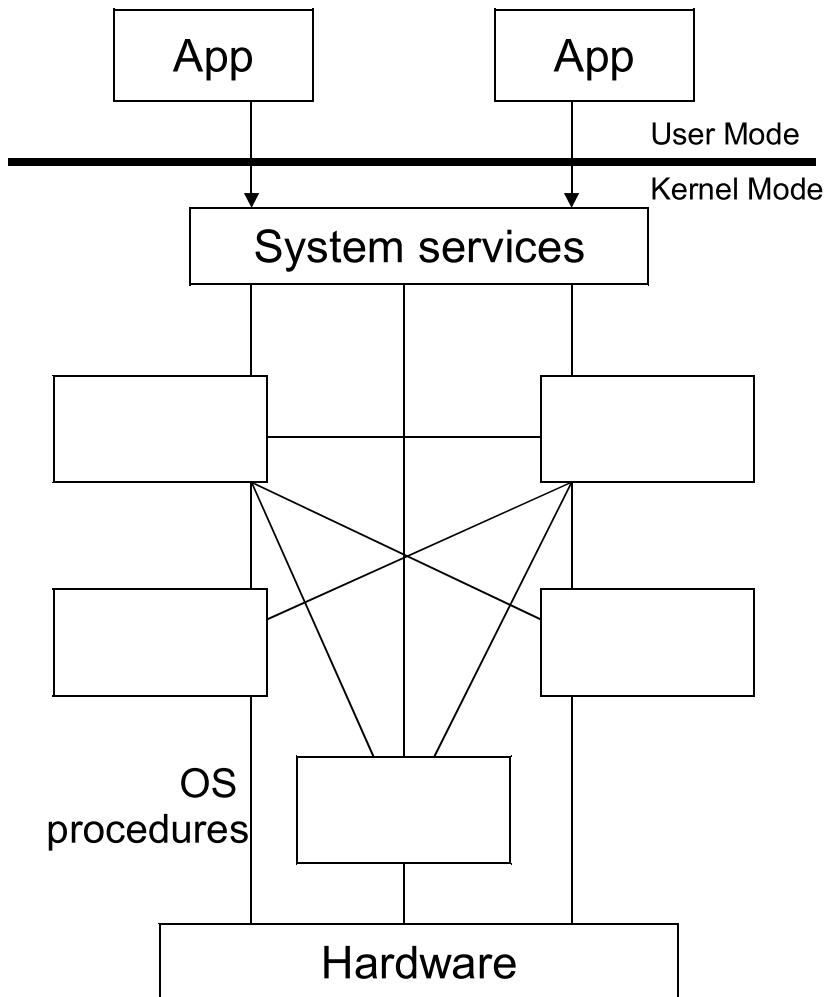


- Module de execuție
 - Kernel mode
 - User mode
- Protecția la memorie
 - Kernel-space
 - User-space



- Context process
 - Cod ce rulează în user-space
 - Cod ce rulează în kernel-space în urma unui apel de sistem efectuat de un anumite proces
- Context întrerupere
 - Codul rulează ca urmare a apariției unei întreruperi
 - Întotdeauna rulează în kernel space

Monolithic vs microkernel



- Linux
- Unix
- Windows
- Mac OS X
- Minix 3.0
- QNX

- Componente selectate la compilare
- Module kernel – încărcare dinamică
- Stratificarea (logică) a codului
- Interfața dintre straturi este hard coupled: macrou-uri, funcții inline, pointeri la funcții
 - Performanțe foarte bune
 - Nu există protecție

Many operating systems and kernel experts have dismissed the label as meaningless, and just marketing: Linus Torvalds said of this issue:

"As to the whole 'hybrid kernel' thing - it's just marketing. It's 'oh, those microkernels had good PR, how can we try to get good PR for our working kernel? Oh, I know, let's use a cool name and try to imply that it has all the PR advantages that that other system has'."

- Spațiul de adresă fizic
 - RAM-ul + periferice
- Spațiu de adresă (virtual)
 - Modul în care procesorul vede memoria: o zonă continuă de memorie de 4GB (32biti)
 - Spațiul de adresă al unui proces
 - Modul în care un proces vede memoria
 - Spațiul de adresă kernel
 - Zona din spațiul de adresă care este rezervat kernelului sau
 - Spațiul de adresă dedicat kernelului

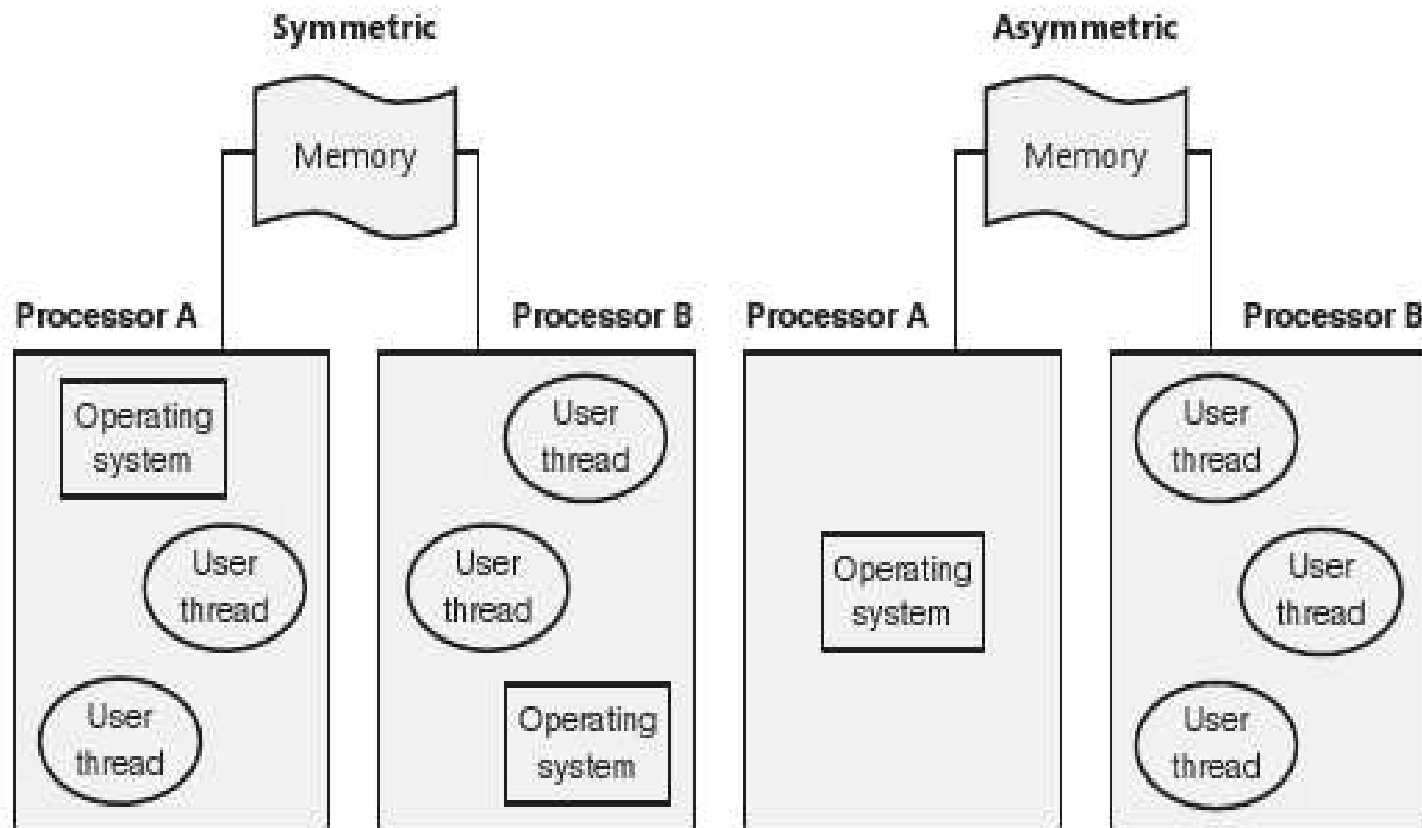
- Sisteme de operare ce permit execuția simultană a mai multor procese
- Mod de implementare
 - Cooperativ
 - Preemptiv

- Un kernel este preemptibil dacă un proces poate fi preemptat chiar și atunci când rulează în kernel.

- Memorie folosită exclusiv de kernel (cod, variabile globale, stivă, memorie alocată dinamic) care poate fi swap-ată.

- Fiecare proces are asociată o stivă kernel, folosită pentru a menține starea procesului pe durata execuție în kernel space
- Dimensiunea stivei este redusă (4KB – 12 KB)

- Există cel puțin trei straturi în kernelele moderne:
 - Cod dependent de arhitectură (C & ASM)
 - Cod independent de arhitectură (C):
 - core
 - device drivere



- Modul în care cresc performanțele relativ la numărul de procesoare din sistem
- Fine grained vs Loose grained locking
- Algoritmi lock free
- Complexitatea algoritmilor

