LUENTO 12

Käyttöjärjestelmät II

Tietoturva - esimerkki KJ:t

UNIX/Linux: Ch 10.7 [Tane 01]

W2000: Ch 11.8 [Tane01], Ch 16.6 [Stal 05]

Distributed Processing

Ch 14 [Stal 05]

UNIX tietoturva

Ch 10.7 [Tane 01]

Unix tietoturva

- **Käyttäjän tunnistus, tiedot PCB:ssä**
  - UID (User ID)
    - kokonaisluku 0-65535
  - GID (Group ID)
- **Tiedostossa vastaavasti**
  - omistaja, joka voi muuttaa oikeuksia
  - oikeudet omistajalle, ryhmälle ja muille
- **Tiedoston käyttö: tarkista onko omistajalla/ryhmällä tarvittavat oikeudet tiedostoon**
  - tarkistus vain tiedoston avaamisen yhteydessä
- **Kaikki KJ oliot ovat ”tiedostoja”**

UNIX käyttöoikeudet

- **Tiedoston attribuutit (i-node)**
  - omistaja (uid), ryhmä (gid)
  - käyttöoikeudet (mode-kentän rwx-bitit)
- **Käyttäjän uid ja gid käyttäjätietokannasta**
  - **`/etc/passwd`** uid ja ensisijainen gid
  - **`/etc/group`** käyttäjän muut ryhmänumerot
- **uid ja gid periytyvät lapsiprosesseille ja edelleen luoduille tiedostoille**
  - voi vaihtaa ohjelmallisesti

UNIX käyttöoikeudet

- **rootilla (uid=0) kaikki oikeudet kaikkeen**
- **Käyttäjien jaottelu**
  - **u** omistaja
  - **g** samaan ryhmään kuuluvat
  - **o** muut käyttäjät
- **Oikeuksien jaottelu u, g, o**
  - **-** ei mitään
  - **r** lukuoikeus
  - **w** kirjoitusoikeus (oikeus muuttaa)
  - **x** suoritusoikeus
- **Uusien tiedostojen käyttöoikeudet PCB:ssä olevan umask-oletuksen mukaan**
  - periytyy rajoitetusti
  - käyttäjän oikeudet, umask, luonnin optiot

UNIX käyttöoikeudet

- **Hakemiston käyttöoikeudet**
  - **r** oikeus listata hakemiston sisältö
  - **w** oikeus poistaa tiedosto hakemistosta
  - **x** oikeus käyttää hakemistonimeä polkunimessä
- **Oikeudet oltava kaikkiin polkunimen osiin**
- **Käyttöoikeuden hetkellinen laajennus, esimerkki:**
  - vain rootilla w-oikeus */etc/passwd* tiedostoon
  - *passwd*-ohjelmalle asetettu **SETUID** bitti
    - **effective userid** on tämän ohjelman (tiedoston passwd) ownerid
  - käyttäjä saa *passwd*-ohjelman suoritusajaksi root-oikeudet (koska root on owner), ja voi muuttaa oman salasanansa
  - **SETGID** bitti vastaavasti (**SETGID** bitti)
    - **effective groupid**

normal

- rw- --- --- 1 root

- srw- s--- t--- 1 root

advanced special permissions

”sticky bit” keep file on swap device

Fig 10-39 [Tane 01]

System call	Description
s = chmod(path, mode)	Change a file's protection mode
s = access(path, mode)	Check access using the real UID and GID
uid = getuid( )	Get the real UID
uid = geteuid( )	Get the effective UID
gid = getgid( )	Get the real GID
gid = getegid( )	Get the effective GID
s = chown(path, owner, group)	Change owner and group
s = setuid(uid)	Set the UID
s = setgid(gid)	Set the GID

Fig. 10-39. Some system calls relating to security. The return code *s* is −1 if an error has occurred; *uid* and *gid* are the UID and GID, respectively. The parameters should be self explanatory.

[Tane 01]

UNIX: Käyttöoikeudet

- **Eräissä järjestelmissä myös käyttäjäkohtaisia pääsylistoja (ACL)**
  - Solaris, HP-UX
    - esim. tietotekniikkaosaston kone “sirppi”
    - man acl
  - Linux
    - ext2:ssa varauduttu toteuttamaan
      - 8 tavua *i-node*:ssa
      - **File ACL** ja **Directory ACL** -kentät

```
setfacl -m u:jussi:r tiedostoX
```

Linux PAM

- **PAM – Pluggable Authentication Module**
- **Parannettu autentikointi, hylkää huonot salasanat, vaadi salasanan vaihtoa aika ajoin**
- **Kerberos optio**
  - keskitetty organisaation turvajärjestelmä
  - käyttäjän tunnistaminen
  - TGS – Ticket Granting Service
    - valtakirjat verkkopalveluihin
    - väärentämättömiä, vain vähän aikaa voimassa olevia valtakirjoja
- **Älykortti- ja äänitunnistus optiot**

Linux ext2fs tiedonsuojaus

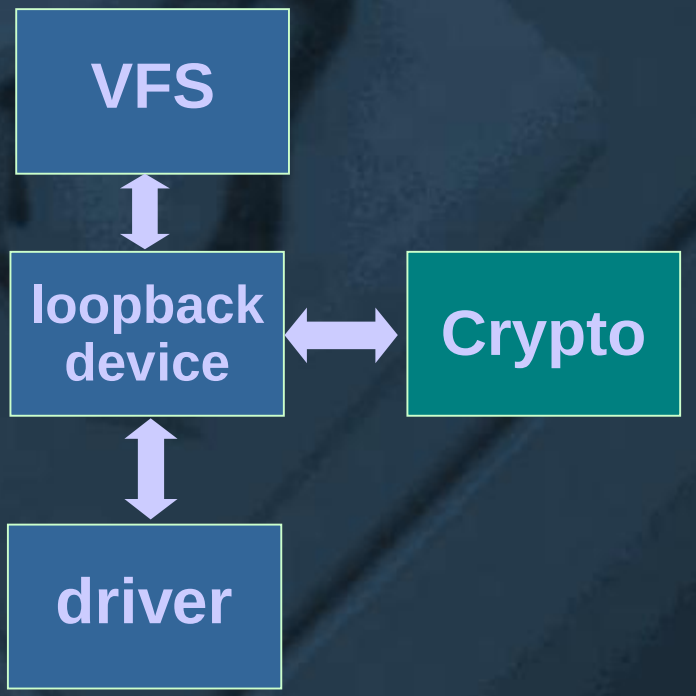
- **Kuten std UNIX**
  - user, group, other
  - r, w, e, x
  - setuid, setgid
- **Tiedostolle myös**
  - a append only
  - i immutable
    - ei voi muuttaa, tuhota tai vaihtaa nimeä
    - ei voi linkittää (hard link, symbolic link)

LSM - Linux Security Module

- **Määrittely ylimääräiselle valvontamoduulille**
  - ladattava ytimen moduuli
  - aktivoituu vasta, kun std pääsynvalvonta on ensin hyväksynyt käyttäjän tai resurssin käytön (LSM on lisäsuoja)
- **LSM SELinux (Security Enhanced Linux)** http://www.nsa.gov/selinux/
  - NSA – National Security Agency (USA)
  - MAC – Mandatory Access Control
    - joka tiedostolle selkeät oikeudet (write up, read down)
    - sääntöjoukko, jota käyttäjät eivät voi manipuloida
  - jäykkä, tehokas, luotettava
- **LSM Capabilities**
  - valtakirjaperustainen pääsynvalvonta
    - i-node:n kentät File ACL ja Directory ACL
  - tarkemmat oikeudet sovellukselle käyttäjästä riippumattomasti
  - POSIX.1e suojausstandardi

Linux kryptografiamoduuli

- Cryptographic API -määrittely
- VFS (virtual file system) ei kutsu laiteajuria suoraan, vaan välissä on **loopback device**
- Loopback device käyttää tarvittaessa kryptomoduulia aina tiedostoa käytettäessä
- per hakemisto?
- per tiedostojärjestelmä?



Käyttöjärjestelmät II

Windows 2000 Tietoturva

W2K Tietoturva

- **Noudattaa ”Orange Book” C2 luokitusta**
  - Dept of Defence (US) Security requirements C2
  - Trusted Computer System Evaluation Criteria
- **C2 – ei kovin paljoa vaadittu**
  - henkilökohtainen kirjautuminen (ei ryhmä)
  - pääsy vain sallittuihin tiedostoihin ja ohjelmiin
- **Muita, parempia turvatasoja**
  - B1, B2, B3
    - B1: kuten C2 ja Mandatory Access Control (MAC)
    - B3: kuten B2 ja kaiken monitorointi ja suojausdomainit
  - A1, A2
    - A1: kuten B, mutta formaalisti todistettu oikein toimivaksi
    - A2: määritellään joskus myöhemmin

http://www.dynamoo.com/orange/summary.htm

click

W2K Suojausympäristö

Fig 16.12 (a) [Stal 05]

- **Joka prosessilla suojauslipuke (access token)** (valtakirja)
  - prosessin tunnistetiedot, ”kuka minä olen”
    - annetaan järjestelmään kirjautumisen yhteydessä
    - omistaja, ryhmä (POSIX)
  - luotaville objekteille määrätyt oletusoikeudet
    - default ACL
  - mahdolliset erityisoikeudet (’special power’)
    - shutdown, write file Y
  - periytyy lapsiprosesseille
  - voidaan muuttaa prosessikohtaisesti
- **Joka oliolla suojauskuvaaja (security descriptor)**
  - suojauskuvaajassa pääsylista
    - discretionary ACL

Fig 16.12 (b,c) [Stal 05]

- **Tarkistus: vertaa prosessin (käyttäjän) pääsylippua olion (kohteen) pääsylistaan**

W2K suojauskuvaaja (security descriptor)

- **Joka oliolla oma suojauskuvaaja**
  - "kuka saa tehdä mitä?"
  - lipukkeita (esim. mitkä kentät käytössä)
  - kohteen omistaja (**owner SID**) tai ryhmä (**group SID**)
    - joku olion luojan suojauslipukkeen SID'eistä
  - **DACL pääsylista** (discretionary access control list)
    - ketkä käyttäjät, mitkä ryhmät saavat käyttää
    - omistaja voi manipuloida
  - **SACL pääsylista** (system ACL)
    - mitä auditiointilokiin, erityisoikeuksien käyttö
    - omistaja ei saa manipuloida (yleensä)

Fig 16.12 (b) [Stal 05]

discretionary = vapaa harkinta, päätösvalta, harkinnan varainen

W2K suojattujen olioiden käyttö

- **Ensimmäinen viite (esim. tiedoston avaus)**
  - vertaa prosessin pääsylippua olion pääsylistaan (DACL)
  - etsi ensimmäinen **ACE** (access control element), joka sopii tähän käyttäjään tälle käyttötavalle
  - jos kaikki kunnossa, anna **kahva** (**handle**, valtakirja) olioon
- **Myöhemmät viitteet kahvan avulla**
  - tarkista aina, että käyttötapa on sellainen, joka oli mukana jo ensimmäisellä kerralla kun pääsy olioon sallittiin
  - jos prosessi yrittää saamansa ”read”-oikeuden asemesta kirjoittaa, niin se ei onnistu
  - jos olion omistaja poistaa ”read” oikeuden, niin se ei estä vanhoja käyttäjiä lukemasta

W2K DACL – Discretionary ACL

- **Koostuu useasta pääsyelementeistä**
  - **ACE** (Access Control Element)
- **Kaksi ACE-tyyppiä**
  - **Allow** – kuka ei saa käyttää ja miten
  - **Deny** – kuka saa käyttää ja miten
- **Käyttö: käy listaa läpi kunnes tälle käyttäjälle (SID) ja käyttötavalle löytyy ensimmäinen ACE ja menettele sen mukaan**
  - sijoita Deny ACE -elementit ennen Allow ACE –elementtejä!
    - esim. kaikki saa, mutta Elvis ei
- **Käyttötavat koodattu pääsyoikeusmaskiin (access mask)**
  - ks. seuraava kalvo (Fig 15.12 [Stal01])

Fig 16.12 (c) [Stal 05]

Fig 11-43 [Tane01]

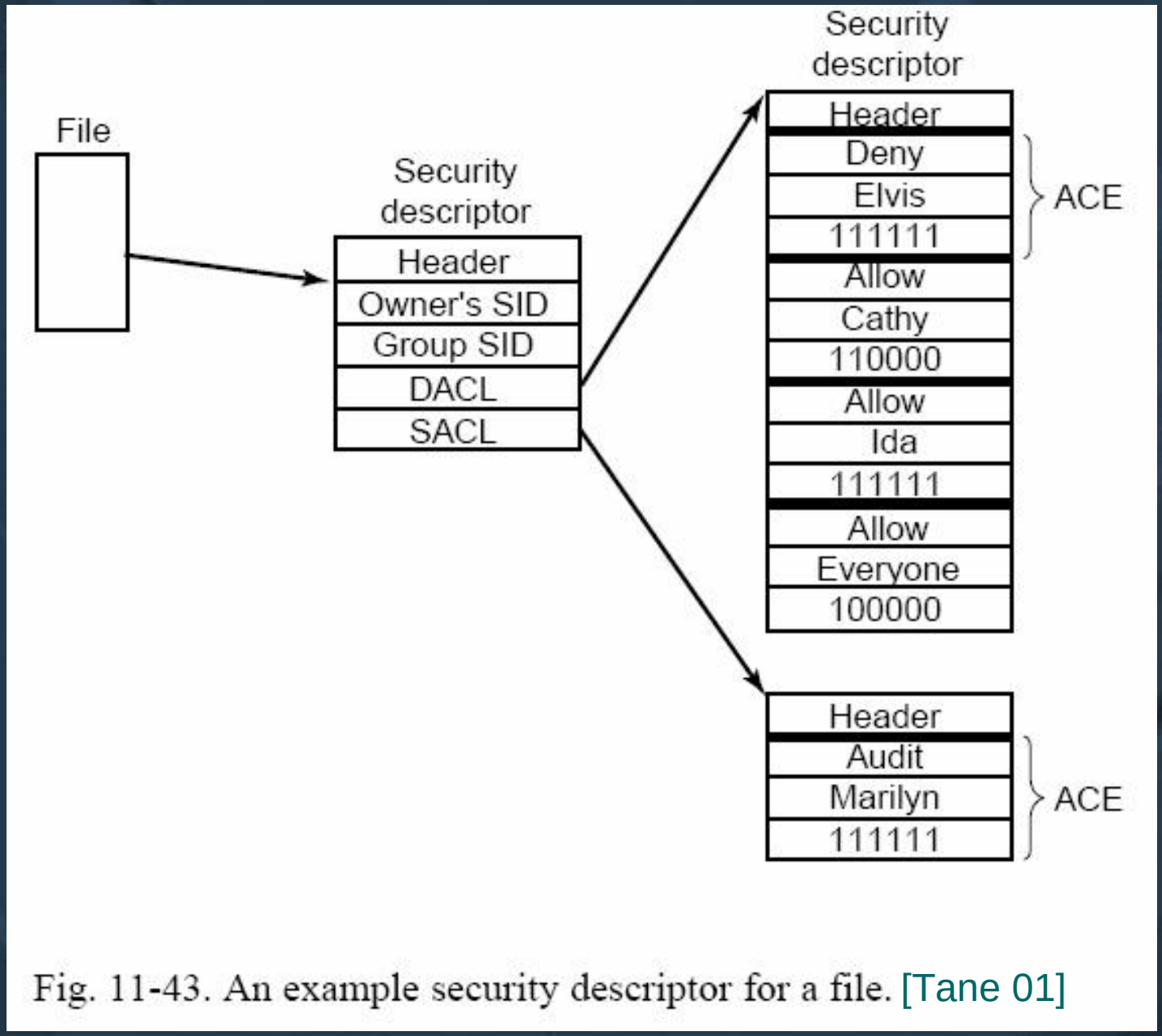
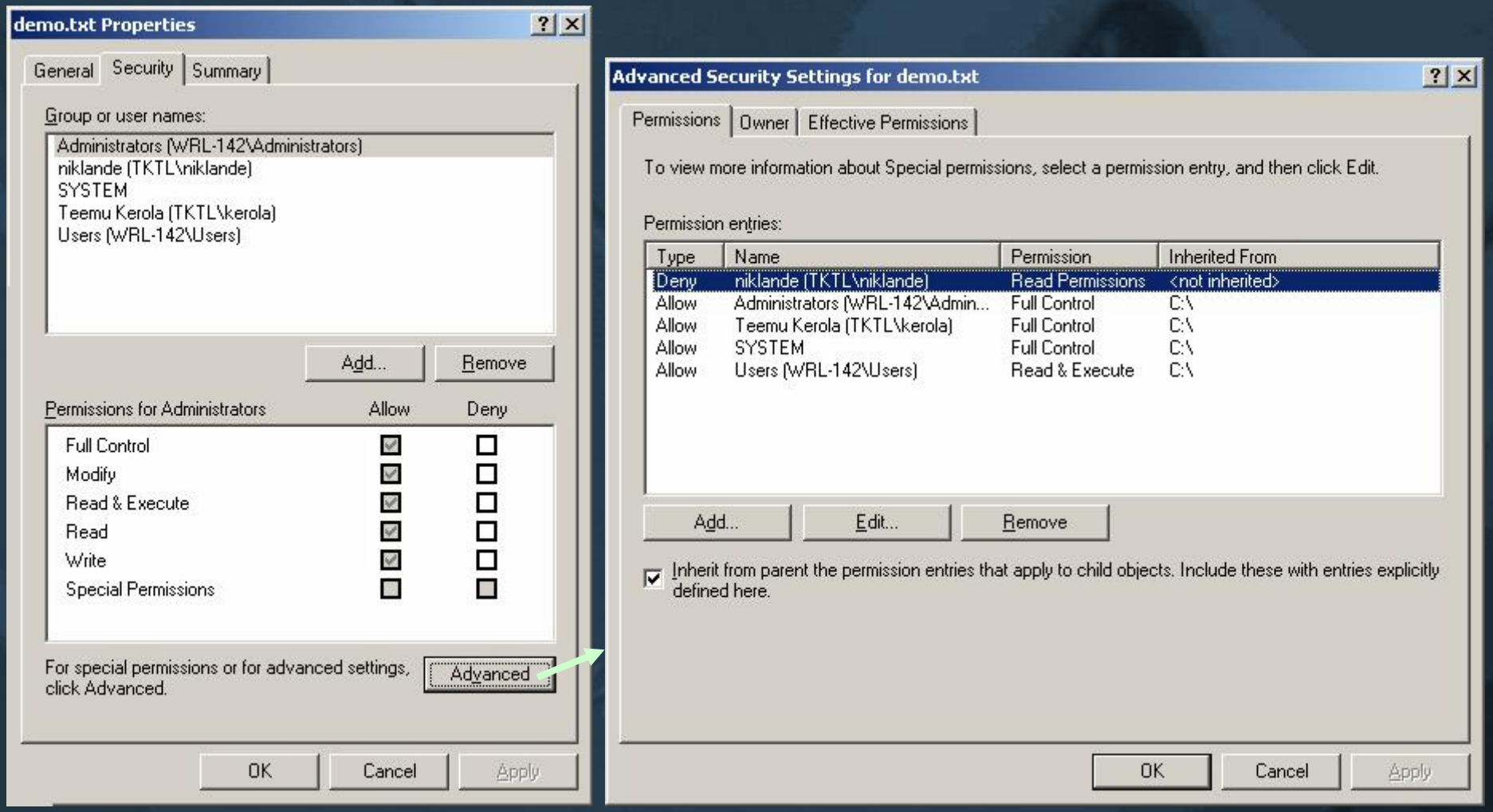
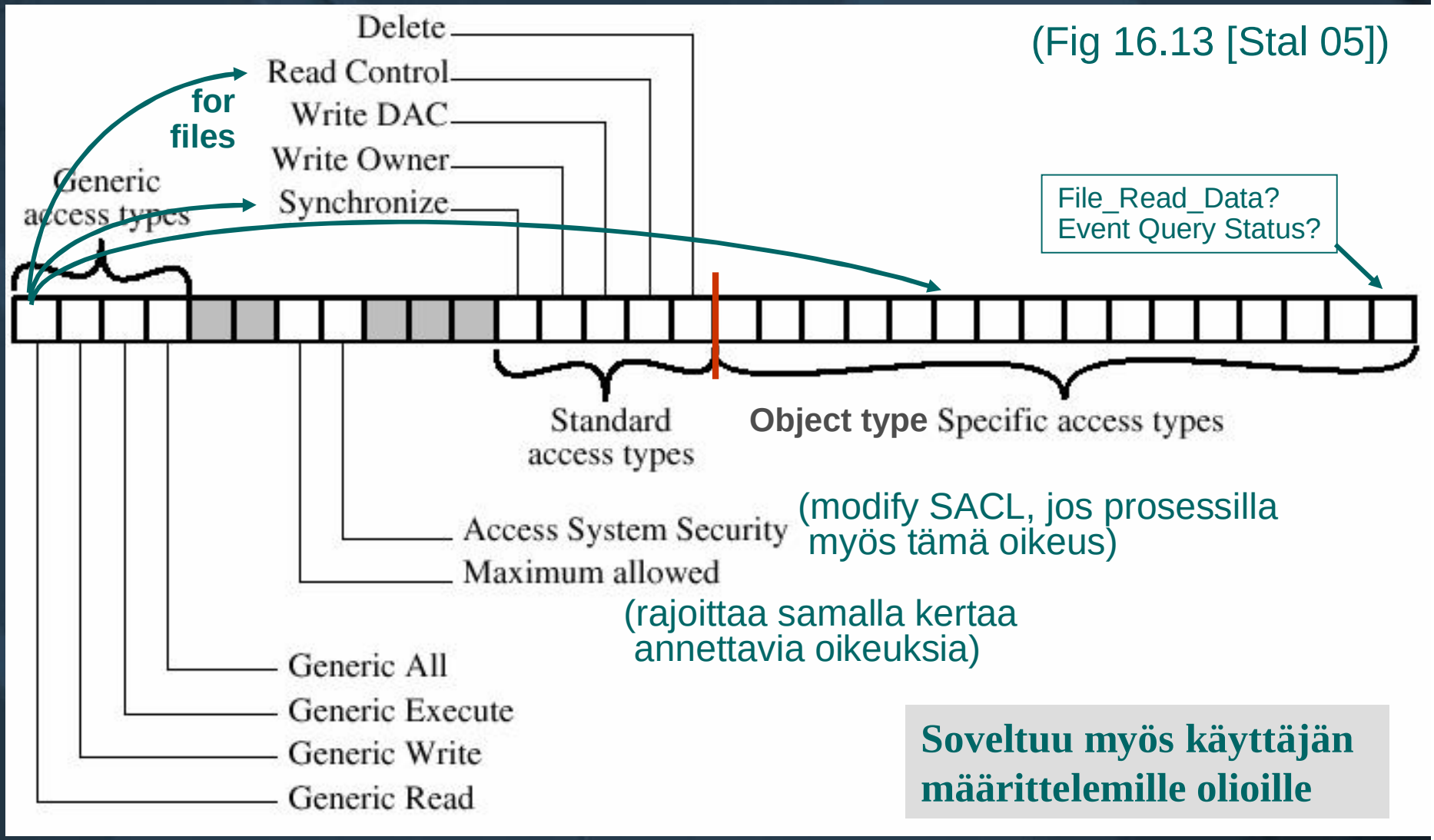


Fig. 11-43. An example security descriptor for a file. [Tane 01]

W2K DACL Esimerkki (NTFS)



W2K pääsyoikeusmaski (access mask)



(Fig 16.13 [Stal 05])

(modify SACL, jos prosessilla myös tämä oikeus)

(rajoittaa samalla kertaa annettavia oikeuksia)

Soveltuu myös käyttäjän määrittelemille olioille

W2K SACL – Security ACL

- **Mistä tapahtumista tähän olioon kerätään auditointilokia**
  - käyttäjä ei tiedä
  - olion omistaja ei tiedä, ei voi muuttaa
- **Esimerkkejä**
  - Marilyn'in kaikki operaatiot tähän olioon pistetään lokiin
  - Kaikkien käyttäjien kaikki operaatiot tähän suojattuun olioon pistetään lokiin

Fig 11-43 [Tane 01]

- **Auditointiloki on olio, jolla oma suojauskuvaaja ja DACL pääsylista**

W2K Security API

Win32 API function	Description
InitializeSecurityDescriptor	Prepare a new security descriptor for use
LookupAccountSid	Look up the SID for a given user name
SetSecurityDescriptorOwner	Enter the owner SID in the security descriptor
SetSecurityDescriptorGroup	Enter a group SID in the security descriptor
InitializeAcl	Initialize a DACL or SACL
AddAccessAllowedAce	Add a new ACE to a DACL or SACL allowing access
AddAccessDeniedAce	Add a new ACE to a DACL or SACL denying access
DeleteAce	Remove an ACE from a DACL or SACL
SetSecurityDescriptorDacl	Attach a DACL to a security descriptor

Fig. 11-44. The principal Win32 API functions for security.[Tane 01]

ACL tarkemmin: Microsoft TechNet artikkeli: click

Operating Systems II

Distributed Processing

Ch 14 [Stal 05]

Distributed Processing

- **Survey of distributed processing capabilities**
  - client-server
  - database applications
  - middleware
  - distributed message passing
  - remote procedure calls
  - clusters

Now:
Ch 14
Oper. Syst. II

- **Distributed Process Management**
  - ”what is in the OS to support distributed processing?”

Later:
Ch 15
separate course
on Distr. Systems
(Hajautetut järj.)

Client/Server

- **Server provides shared services**
  - database server
  - name server
  - web server
  - password server

Fig 14.1 [Stal 05]

- **Access through network (LAN, WAN, Internet)**

Fig 14.2 [Stal 05]

- **Server may also be a client**
- **Database server**
  - database layer below application layer

Fig 14.3 [Stal 05]

Client/Server Application Classes

- **Where is processing done? What part?** Fig 14.5 [Stal 05]
  - Host-based
    - E.g., stupid terminal, not really a client
  - Server-based
    - E.g., web browsing
  - Cooperative processing
    - E.g., general database application
  - Client-based
    - E.g., web browsing with applets
- **Which class best for this application?**
- **What OS support is available?**

Middleware

- **What if client does not know who the server is?**
  - ”I just want this type of service”
- **Clearinghouse for service requests: middleware**
  - uniform access to many resources
  - platform independent
    - OS: Unix, Linux, SVR4, W2000
    - database: Oracle, Gupta
    - DECnet, Novell, TCP/IP

Fig 14.6 [Stal 05]

Fig 14.8 [Stal 05]

Fig 14.9 [Stal 05]

Fig 13.10 [Stal01]

Distributed Message Passing

- **Plain messages for client/server**
  - reliable or not? blocking or not?
- **RPC - Remote Procedure Call**
  - use just like local procedure calls
  - standardized interface
  - reusable modules
  - parameter problems
    - marshalling
    - pointers – call-by-reference
  - non-persistent/persistent binding
    - save handle for remote process or not?
  - synchronous/asynchronous (to block or not)
- **RMI – Remote Method Invocation**
  - for Java users

Fig 14.10 (a) [Stal 05]

Fig 14.11 [Stal 05]

Fig 14.10 (b) [Stal 05]

Fig 14.12 [Stal 05]

Object Oriented Mechanisms

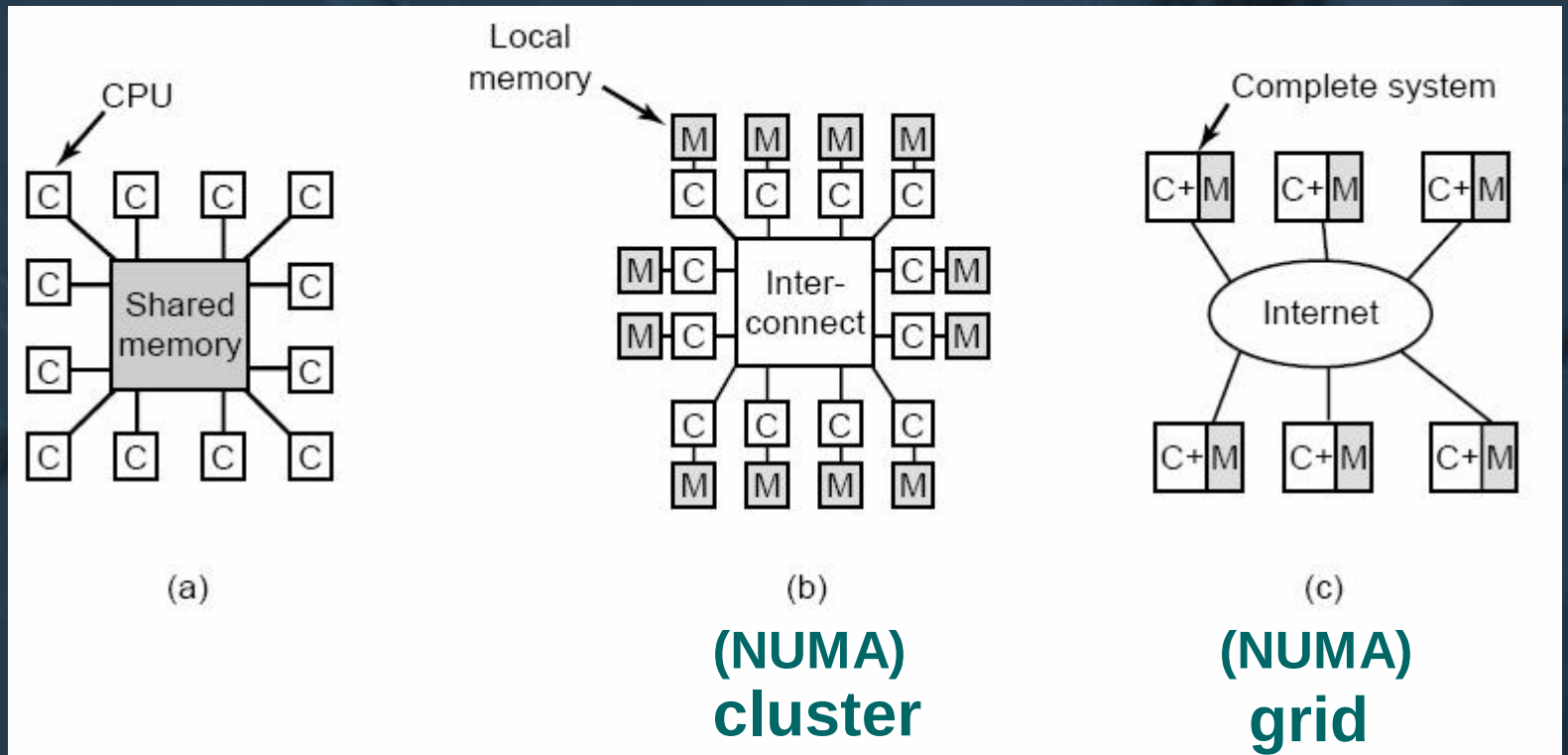
- **ORB – Object Request Broker**
  - higher level concept than RPC or RMI

Fig 14.10 (c) [Stal 05]

- **The good thing about standards is that you can choose which one to use**
  - **DCOM** – Distributed Component Object Model
    - Microsoft, Digital
    - each object can have multiple interfaces
      - interface must be defined when requesting service
  - **CORBA** – Common Object Request Broker Architecture
    - OMG - Object Management Group (non-profit)
    - IBM, Apple, Sun, …
    - **ORB** (Object Request Broker) to ORB communication
    - **IDL** (Interface Definition Language) for programming language independent interface definition
      - one interface per object

Cluster Computer

Shared memory multicomputer



(NUMA) cluster

(NUMA) grid

Fig. 8-1. (a) A shared-memory multiprocessor. (b) A message-passing multicomputer. (c) A wide area distributed system.[Tane 01]

Cluster

- **Cluster, multicomputer, COWS (cluster of workstations)**
- **Many whole (similar?) computers**
  - can work independently if needed
- **Interconnected**
- **Work together**
- **Unified computing resource**
  - e.g. memory, disk
- **Illusion of one machine**

Benefits of Clustering

- **Absolute (?) scalability**
- **Incremental scalability**
- **High availability**
- **Superior price/performance**
  - as compared to what? SMP? Grid? Supercomputer?
- **Disadvantages?**
  - more complexity than uniprocessing or SMP
    - E.g., synchronization
  - communication delay vs. memory access
  - which applications suitable for it?

Cluster Configurations

- **Shared disk or not?**

Fig 14.13 [Stal 05]

- **Passive standby**
  - you would call this ”clustering”?
    - need many whole computers

Tbl 14.2 [Stal 05]

- **Active secondary**
  - separate servers
    - each has its own disks
  - servers connected to disks – ”shared nothing”
    - shared disks, disks partitioned to servers
    - each disk has one ”owner” (user)
  - servers share disks
    - shared disks
    - need mutex locks

Cluster Failure Management

- **High Performance Cluster**
  - no redundancy, just lots of processing power
  - example: Magnetic Resource Image (MRI) scanner
- **Highly Available Cluster**
  - probably all resources available
    - some resources serve as backups
  - no guarantee of transaction execution
  - application provides for consistency
  - example: soft real time
- **Fault Tolerant Cluster**
  - guarantees that all resources available
    - HW redundancy, transaction logging
  - application does not need to provide consistency
  - trouble at resource X?
    - start using alternative (spare) resource **failover** (varalaite käyttöön)
    - repair X or replace X
    - return to using X **failback** (laite takaisin käyttöön)
  - examples: hard real time, aircraft control system

Load Balancing Cluster

- **Incremental scalability**
  - automatic use of new resources
- **Migrate services/work from one computer to another**
  - how to migrate processes?
    - code, data, PCB?
- **Load balancer node**
  - one node dedicated to load balancing
- **Example**
  - e-business with high user volumes

Cluster Application Concurrency

- **Must have application level concurrency**
  - middleware layer to enable co-operation
  - how to find it?
- **Parallelizing compiler**
  - compiler does the parallelization work
    - ”dusty decks” OK, though may not be so good
  - may make compiled application dependent on cluster size
- **Parallelized application**
  - programmer does the parallelization work
    - hard work, complex
  - may make application really dependent on cluster size
- **Parametric computing (parallelized problem)**
  - run many instances of same application, one in each node, with different parameters
    - simple, but not suitable so often

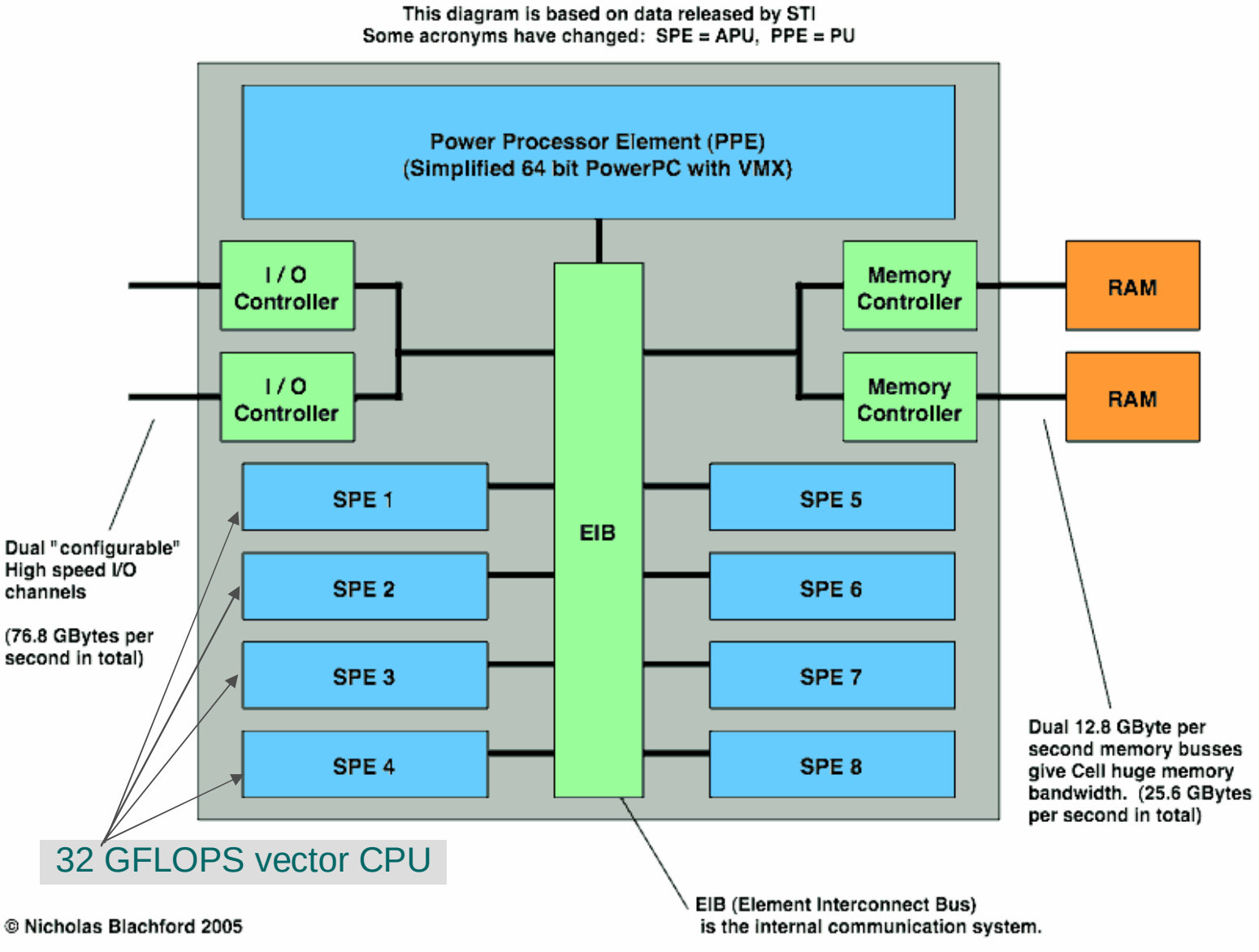
Fig 14.14 [Stal 05]

Cluster Middleware

Fig 14.14 [Stal 05]

- **Single everything – feels like one computer**
  - system image, entry point, control point
  - virtual networking
  - memory space
  - job management
  - user interface
  - I/O space
  - process space
- **Checkpointing**
  - failure recovery
- **Process migration**
  - load balancing

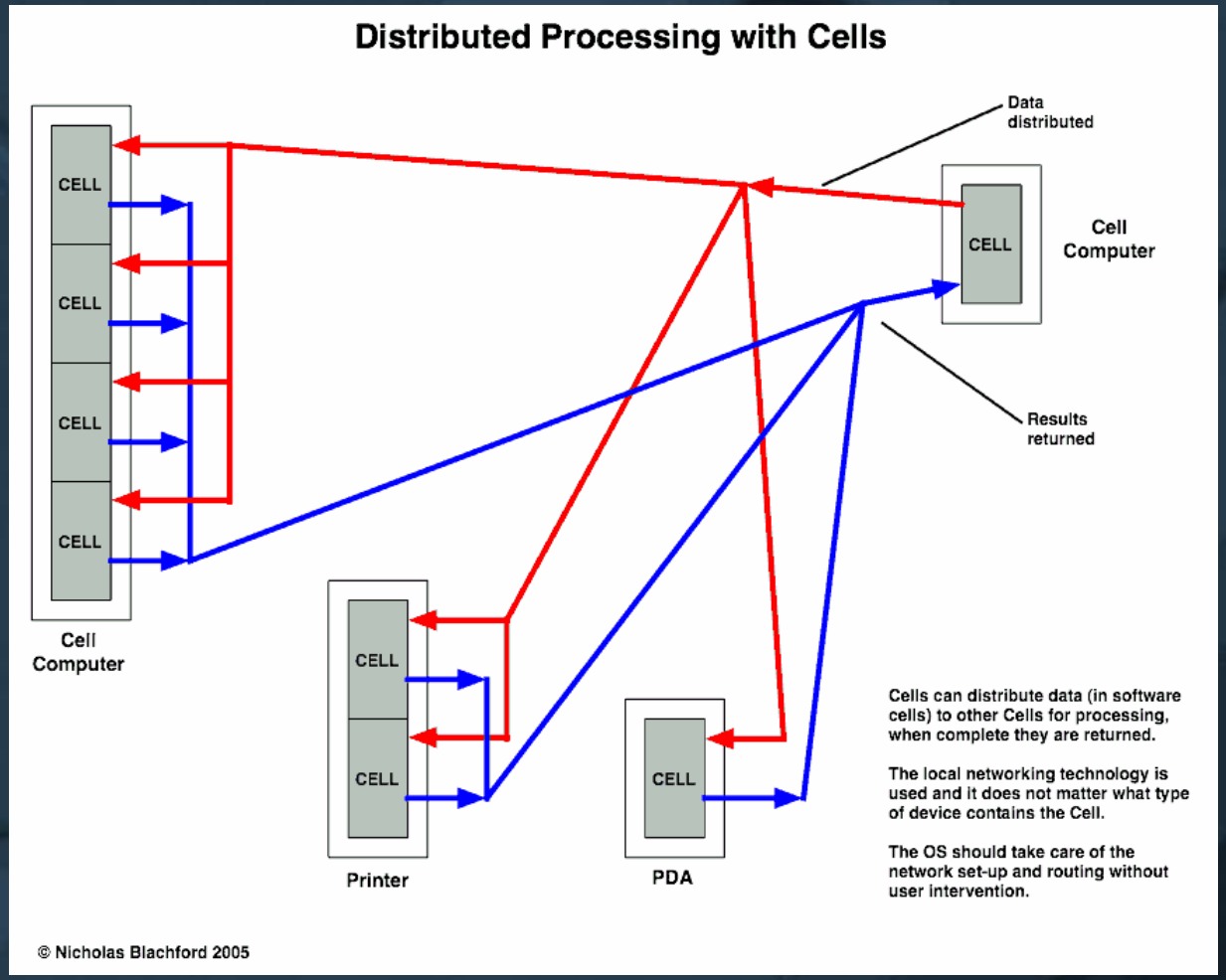
Sony/Toshiba/IBM Cell Processor Architecture



http://www.blachford.info/computer/Cells/Cell1.html

Cell Operating System

- Cell OS?
- Load balancing?
- Migration?
- Shared mem?
- Parallelizing compiler?
- Shared memory cluster?
- Distributed shared memory cluster?



http://www.blachford.info/computer/Cells/Cell1.html

W2000 Cluster Server (Wolfpack)

Fig 14.15 [Stal 05]

- **Shared nothing**
  - shared disk, each disk volume has one owner/user
  - max 32 nodes, max 32 GB memory
- **Cluster service (cluster middleware)**
  - at each node
- **Cluster node *resource***
  - disk drive, network card, application, database, TCP/IP address, …
  - ”***online***”, if resource available to others
  - packaged into ***groups***
    - e.g., all resources needed to run one application
    - unit of failover and load balancing
- **(New and better: W2003 Cluster Server)**

W2000 Cluster Service (contd)

- **Middleware layer**
- **Node manager**
  - who is in cluster now?
  - hearbeat messages to other node managers
  - no heartbeat from node X for a while ⟶ X is dead!
- **Configuration database manager**
  - who owns what resources
  - fault-tolerant transactions
- **Resource manager & failover manager**
  - startup, reset, failover
- **Event processor**
  - cluster components synchronize with events

Fig 14.15 [Stal 05]

Beowulf Cluster with Linux

Fig 14.18 [Stal 05]

- **Beowulf 1994**
  - are many cheap PC's better than one good workstation?
  - yes….
- **Beowulf features**
  - normal cheap components, no custom components, many vendors
  - dedicated processors, dedicated network
  - one controlling node (**front end node**, or **head node**)
  - similar slave computers (for easy load balancing)
  - scalable I/O
  - freely available software
  - freely available distribution computing tools
  - give design and improvements to the community (free?)
- **Examples**
  - ETH Zurich, 251 nodes, 502 processors (June 2001)
  - Niflheim Linux cluster, 5.0-TeraFLOPS, 945 node supercomputer
- **Beowulf Cluster with Windows**

Beowulf Software

Fig 14.18 [Stal 05]

- **Each node has own copy of Linux kernel**
- **Autonomous Linux system**
- **Kernel extensions to participate in global namespaces**
  - cluster middleware
  - Beowulf Distributed Process Space (BPROC)
    - start remote processes without login
    - remote processes visible in cluster *front end* node
  - Beowulf Ethernet Channel Bonding
    - load balancing over multiple Ethernets
    - LAN, not WAN, not internet
  - Pvmsync
    - distributed synchronization within cluster
  - EnFuzion
    - tools for parametric computing
      - control jobs in remote nodes

Grid Computing

- **Utilize idle computing resources in Web**
  - home computers?
  - company computers?
- **Many layers to utilize heterogeneous computers**
  - **application layer**
  - **collective layer** for coordination
  - **resources layer** for sharing resources
  - **connectivity layer** for connections
  - **fabric layer** for physical resource usage
- **Examples**
  - SETI@home click
  - Globus toolkit for business solutions (Globus Alliance) click

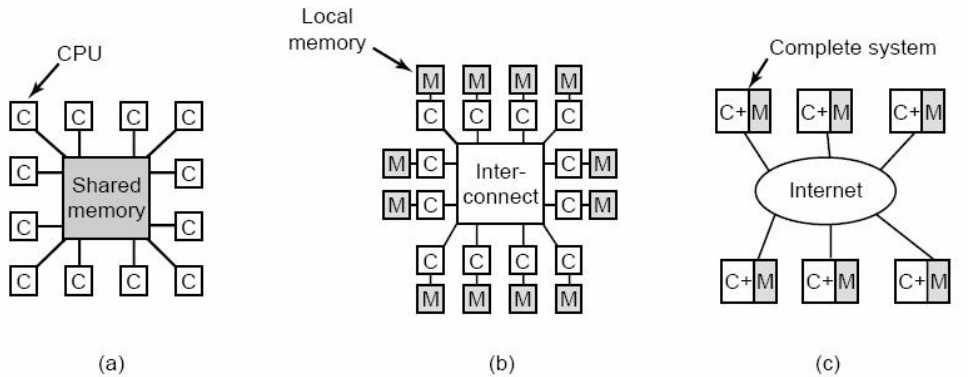


Fig. 8-1. (a) A shared-memory multiprocessor. (b) A message-passing multicomputer. (c) A wide area distributed system.

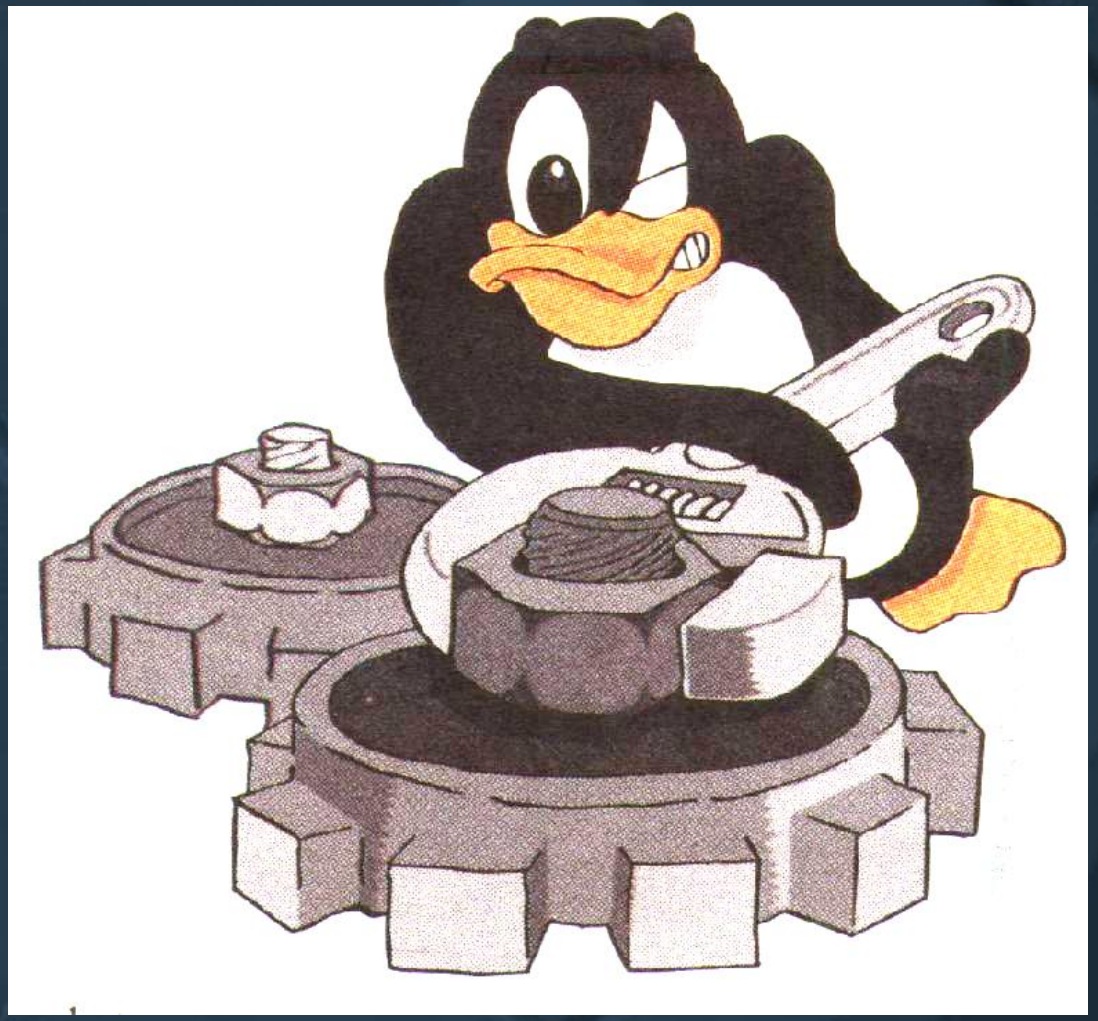
[Tane 01]

Review Questions

- How do Linux and W2000 security features differ?
- What is good/bad with Linux/W2000 security?
- What can be done with Linux but not in W2000?
- What can be done with W2000 but not in Linux?

- What is needed from OS to support clusters?
- What is needed from OS to support grids?
- What synchronizations primitives can (not) be used with clusters?
- What synchronizations primitives can (not) be used with grids?

-- END --



Operating Systems II