



Sistemas Operativos Fisop 2022

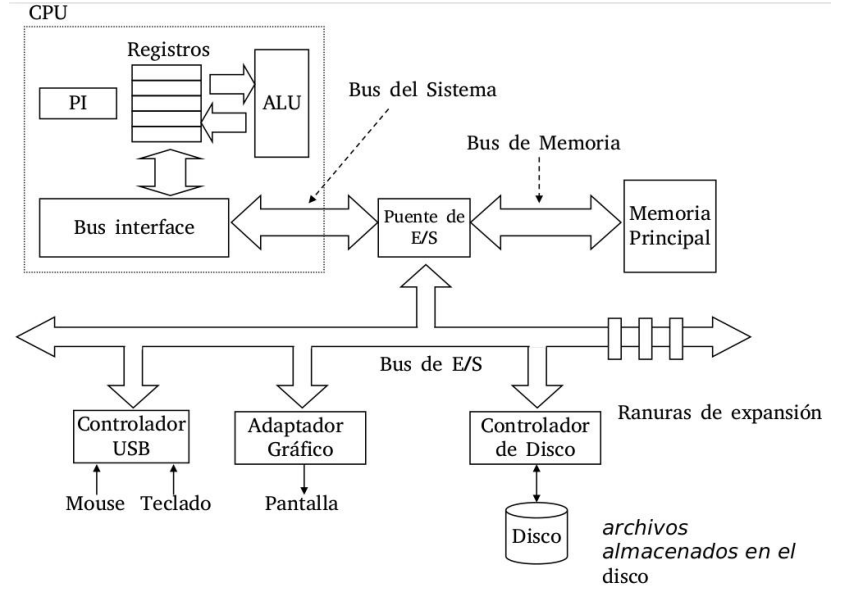
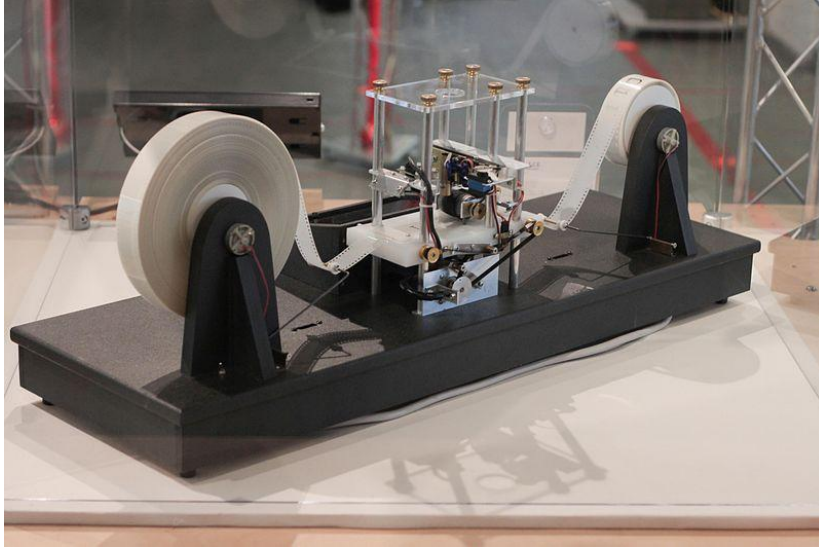
Introducción



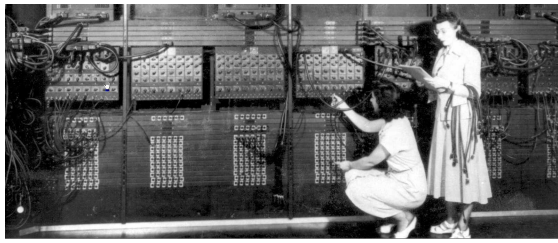
Alan Turing



John Von Neuman



¿Qué es esto?



Eniac



IBM 7094



IBM 370



Baby Manchester



IBM 360



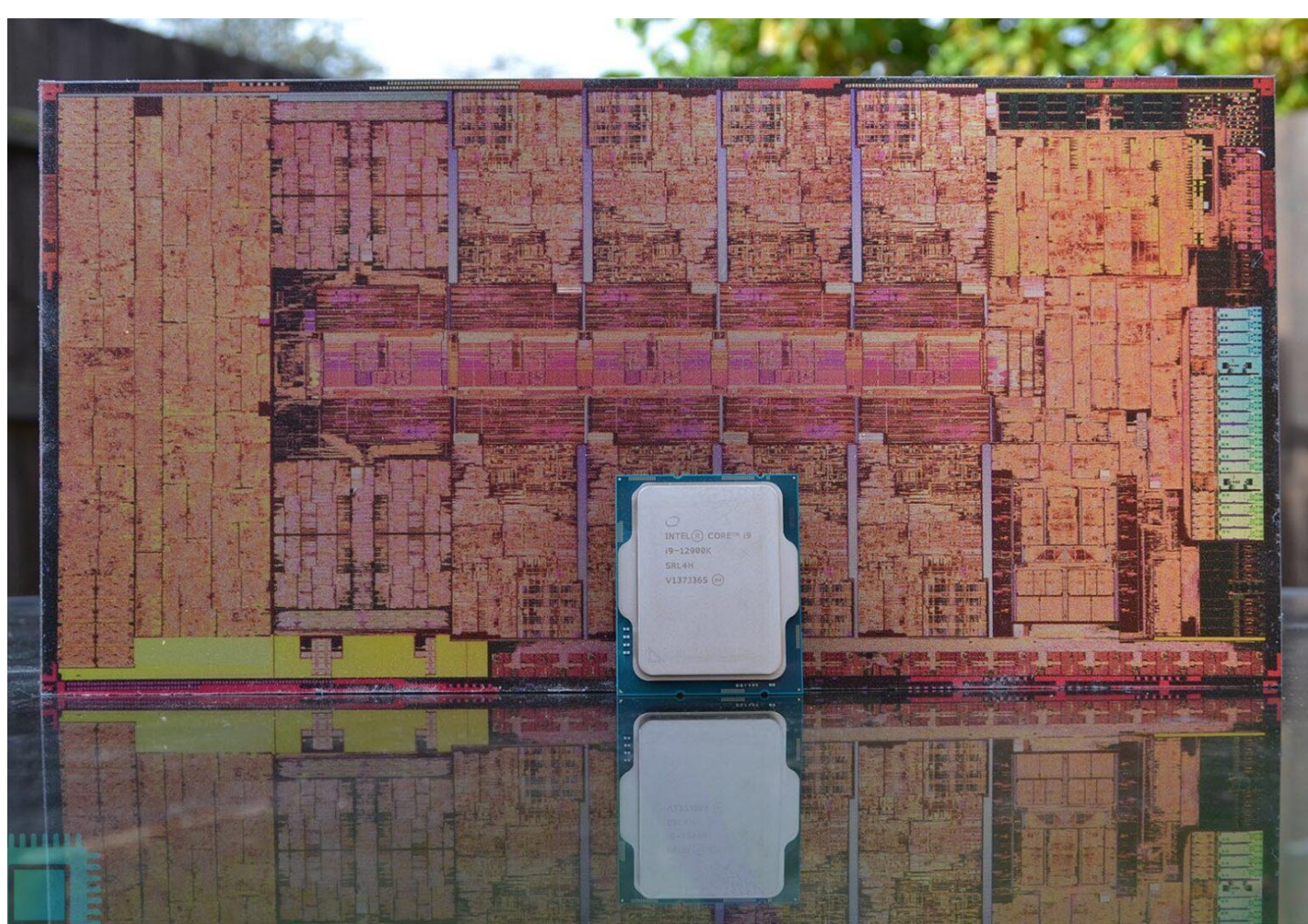
IBM 1401



IBM 390

12th Generation Intel® Core™ i9 Processors

... 8 "Golden Cove"
P-cores and 8
"Gracemont" E-cores.
The E-cores are spread
across two 4-core
"E-core Clusters."



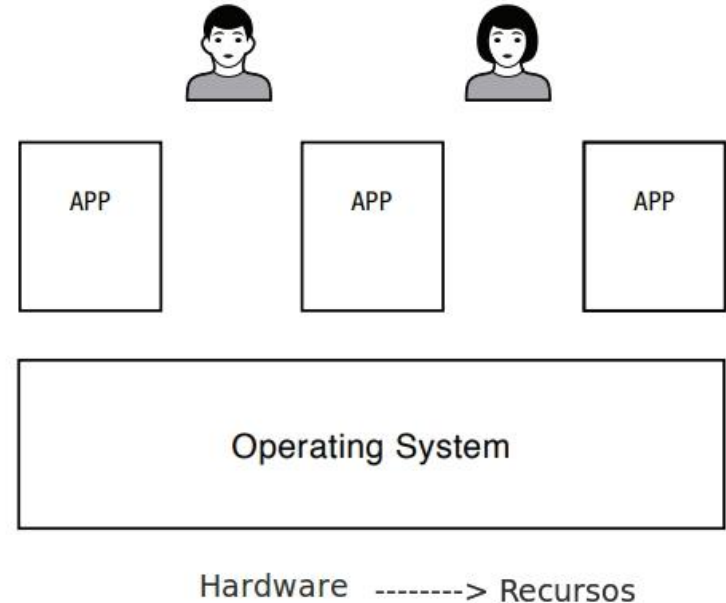


¿Qué es un sistema Operativo?

Un **Sistema Operativo (OS)** es la **capa de software** que **maneja los recursos de una computadora** para sus **usuarios** y sus **aplicaciones**. [DAH]

¿Qué es un sistema Operativo?

En un sistema operativo de propósito general, los **usuarios** interactúan con **aplicaciones**, estas aplicaciones se ejecutan en un **ambiente que es proporcionado** por el sistema operativo. A su vez el sistema operativo hace de **mediador para tener acceso al hardware del equipo**.





¿Qué es un sistema Operativo?

Un sistema operativo es el software encargado de hacer que la ejecución de los programas parezca algo fácil. La forma principal para llegar a lograr esto es mediante el concepto de virtualización. esto es, el sistema operativo toma un recurso físico (la memoria, el procesador, un disco) y lo transforma en algo virtual más general, poderoso, fácil de usar.



Referee

Un OS gestiona recursos compartidos entre diferentes aplicaciones, que se encuentran ejecutándose en la misma máquina física.

- un OS puede frenar la ejecución de una aplicación e iniciar la ejecución de otra.
- los OS aíslan a cada aplicación de las demás que se encuentran corriendo en la misma computadora.
- Por ende un OS tiene que protegerse a sí mismo y a las demás aplicaciones que se están ejecutando en la misma computadora.
- Y dado que todas estas aplicaciones comparten los mismos recursos físicos el OS decide que aplicación usa un determinado recurso y cuando.



Ilusionista

- Un OS debe proveer una abstracción del hardware para simplificar el diseño de aplicaciones.
- Imaginen lo complejo que sería escribir el clásico hola mundo no se tiene que pensar en qué lugar de la memoria física este se encuentra almacenado, ni como esta memoria se comparte con otros datos y aplicaciones.
- El sistema operativo provee la ilusión de que se dispone de toda la memoria para almacenar al programa, cuando realmente se sabe que la memoria ppal de la computadora es finita.



Pegamento

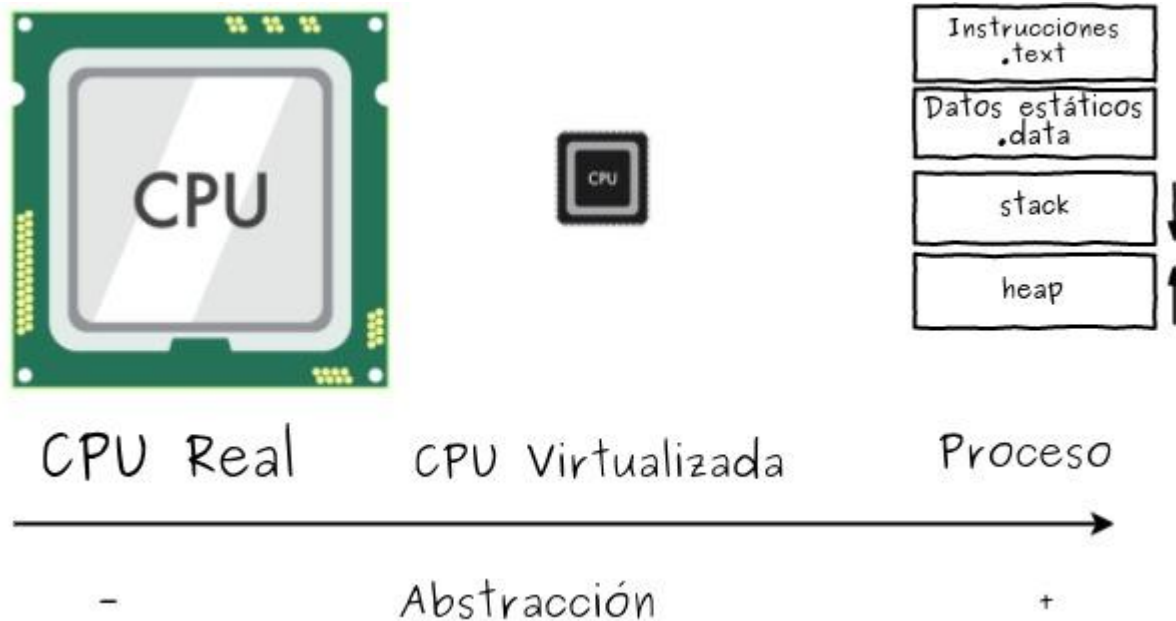
- Un OS debe proveer una serie de servicios comunes que faciliten un mecanismo que permita compartir, por ejemplo, información entre las aplicaciones.... “Cut & Paste” por ejemplo ... este mecanismo es uniforme en todo el sistema.
- Otro ejemplo es el “look and feel” de la interfaz de usuario.
- Tal vez uno de los más importantes sea el mecanismo de acceso a los dispositivos de entrada y salida del sistema, de forma tal que las aplicaciones puedan usarlos independientemente de la marca y modelo de los mismos.

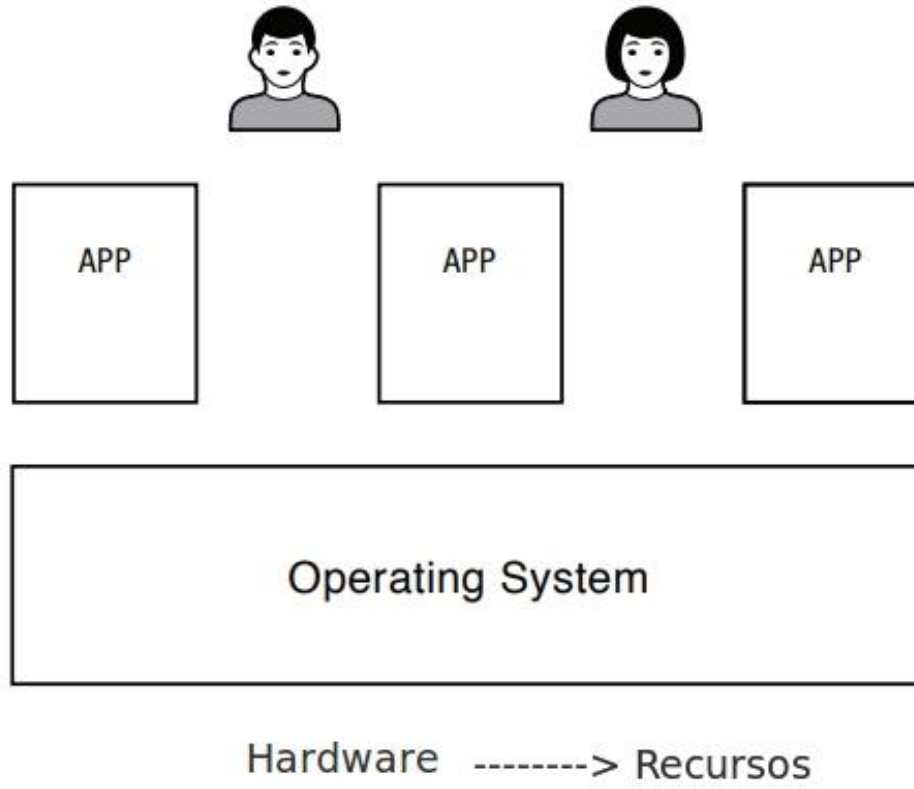


Pegamento

- API general
- Library Call
- System Call

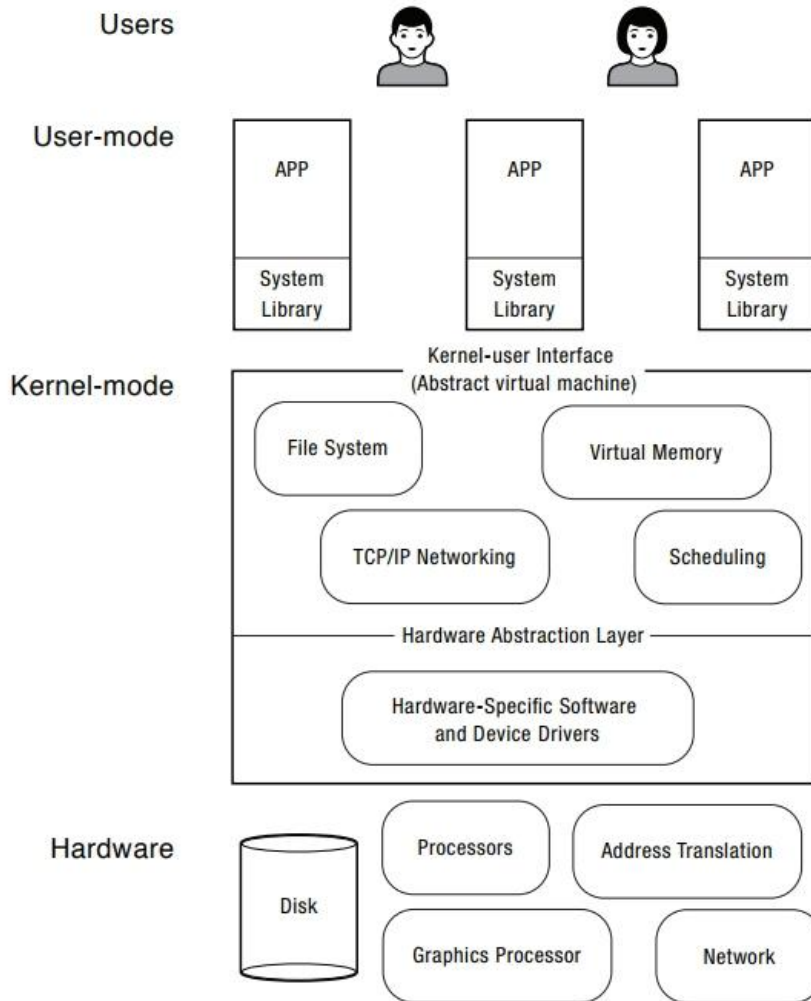
Virtualización





Sistema Operativo

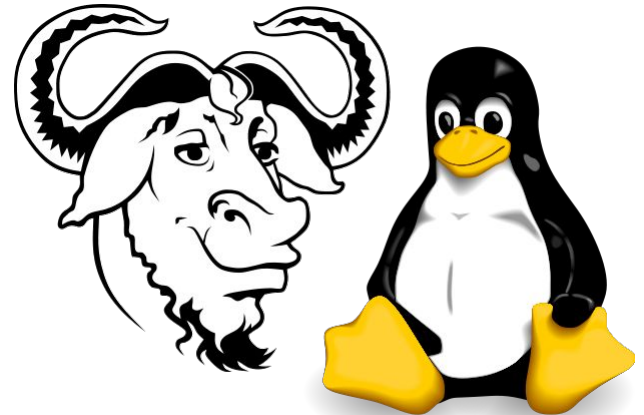
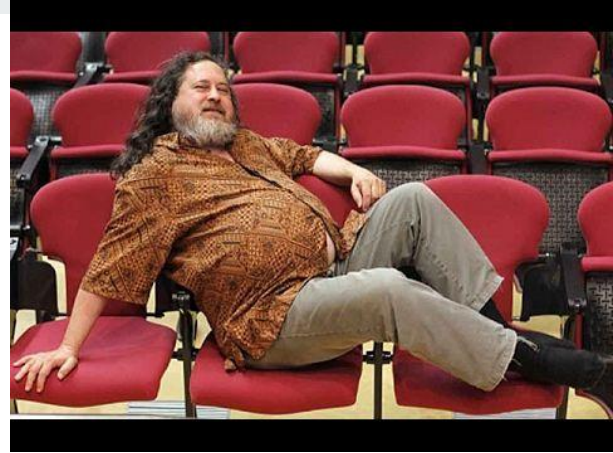
Sistema Operativo



GNU / Linux

Sistema Operativo

<https://directory.fsf.org/wiki/GNU>

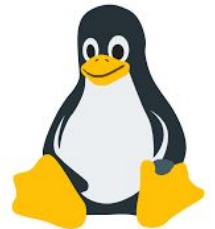




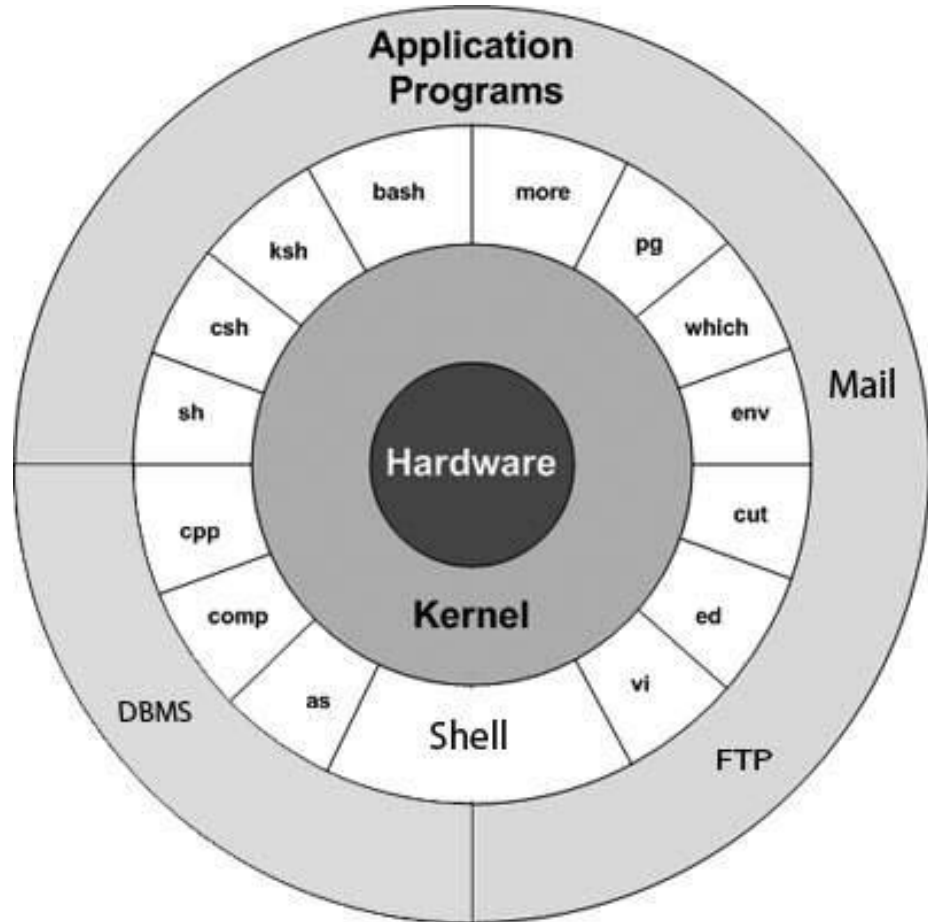
Linux: el Kernel



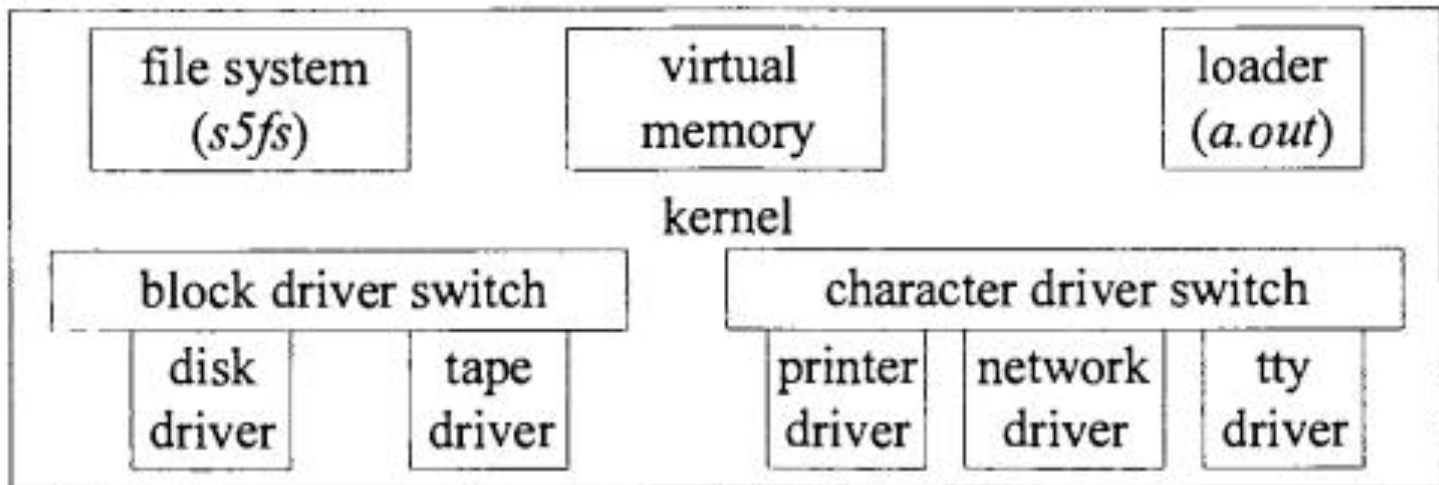
- La capa para la gestión de dispositivos específico
- y una serie de servicios para la gestión de dispositivos agnósticos del hardware que son utilizados por las aplicaciones.



LINUX -> KERNEL

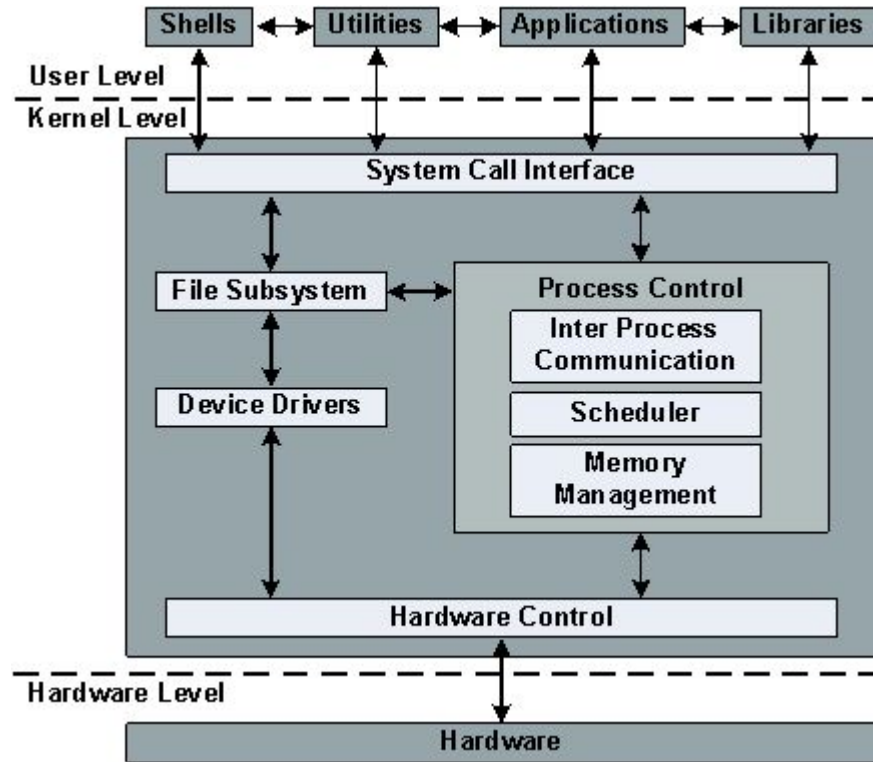


El Kernel



El Kernel

Kernel-land





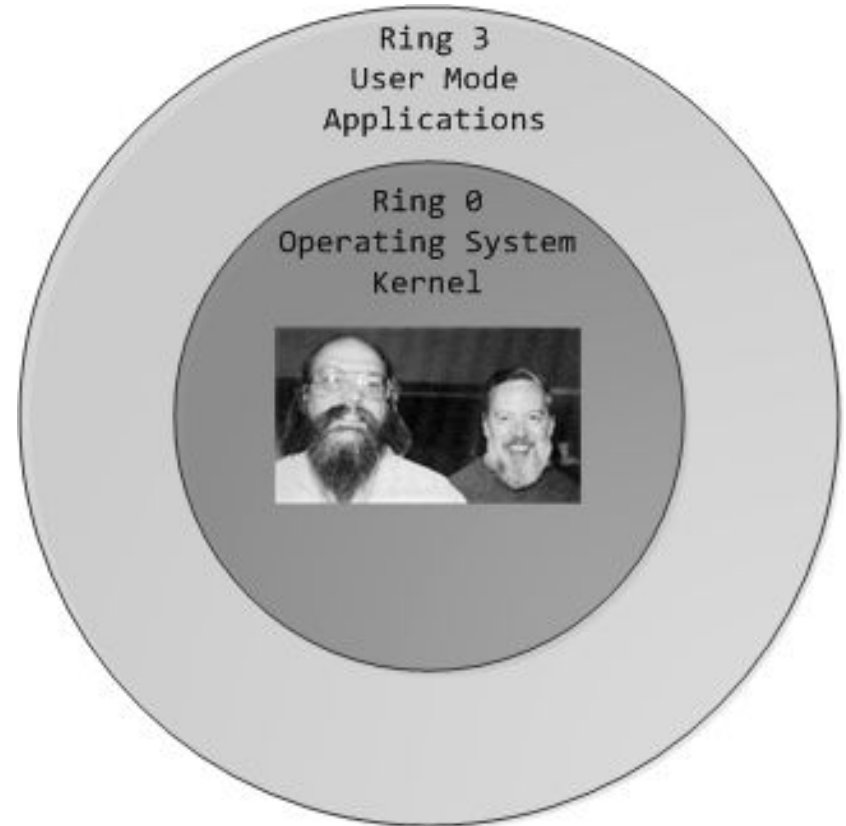
¿Y que pasa en User Land?

Los programas se están ejecutando!

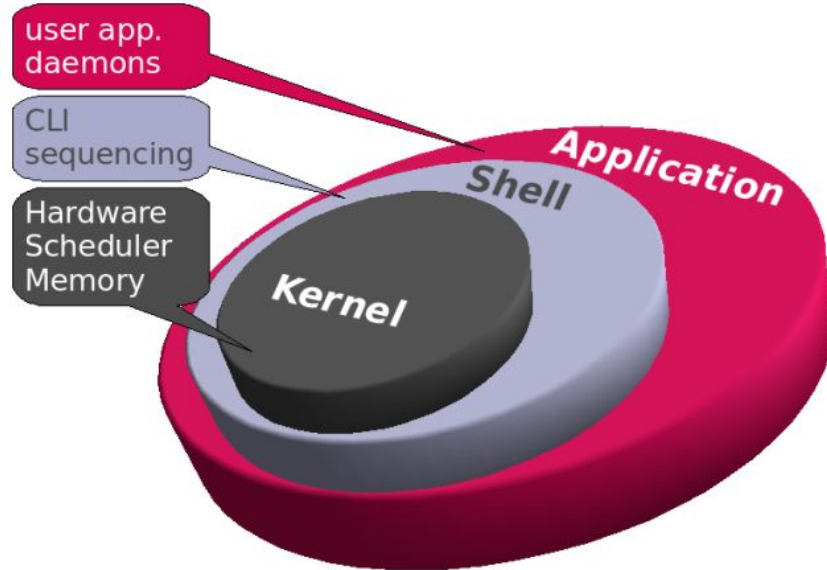
Las aplicaciones se ejecutan en un **contexto aislado, protegido y restringido** y mediante el uso de funciones que se encuentran **bibliotecas pueden utilizar los servicios de acceso al hardware o recursos que el kernel proporciona**. El contexto de ejecución de las aplicaciones se denomina User Mode o modo usuario, mas restrictivo, aislado y controlado.



User-land vs Kernel land



User-land vs Kernel land



La Shell

```
Processes: 123 total, 3 running, 120 sleeping, 55% threads          08:36:09
Load Avg: 1.75, 1.53, 1.49 CPU usage: 25.86% user, 22.83% sys, 55.62% idle
SharedLibs: 3908K resident, 5760K data, 0B linkedit.
MemRegions: 44714 total, 3360M resident, 77M private, 1110M shared.
PhysMem: 921M wired, 5280M active, 758M inactive, 6940M used, 1243M free.
VM: 238G vsize, 1034M framework vsize, 4797528(2) pageins, 0(0) pageouts.
Networks: packets: 581628/454M in, 462610/68M out.
Disks: 229509/3409M read, 418661/7924M written.
```

PID	COMMAND	%CPU	TIME	#TH	#WD	#PDR	#MREG	RPRVT	RSHRD	RSIZE
1477	top	12.9	00:01.38	1/1	0	24	33	1420K+	244K	1998K+
1466-	cvmsComp_i38	0.0	00:00.04	1	0	18	36	1116K	9520K	5760K
1463	bash	0.0	00:00.00	1	0	17	25	296K	856K	968K
1462	login	0.0	00:00.01	1	0	22	62	616K	3200K	2448K
1459	cvmsComp_x86	0.0	00:00.03	1	0	18	34	1592K	9520K	6220K
1456-	Cathode	80.77	00:10.88	5	2	127	267	20M+	98M+	65M+
1454	launchd	0.0	00:00.00	2	0	37	46	236K	428K	660K
1452	quicklookd	0.0	00:00.48	6	2	88-	155	21M-	17M	58M-
1451	ocspd	0.0	00:00.01	2	0	42	40	736K	3192K	2152K
1450	mdworker	0.0	00:00.06	3	1	48	67	1636K	16M	4284K
1294-	Google Chrom	0.3	00:42.07	4	1	93	778	48M	89M	80M
1267-	DashboardCli	0.0	00:01.27	5	2	128	228	14M	26M	21M
1266	DashboardCli	0.0	00:02.39	5	2	129	330	40M	43M	97M
1192-	Google Chrom	0.8	00:10.10	4	1	93	348	19M-	87M	43M-
1014	dd	0.0	00:00.00	1	0	14	23	180K	240K	436K

```
mars@marsmain ~ $ pwd
/home/mars
mars@marsmain ~ $ cd /usr/portage/app-shells/bash
mars@marsmain /usr/portage/app-shells/bash $ ls -al
total 130
drwxr-xr-x 3 portage portage 1024 Jul 25 10:06 .
drwxr-xr-x 33 portage portage 1024 Aug 7 22:39 ..
-rw-r--r-- 1 root root 35888 Jul 25 10:06 ChangeLog
-rw-r--r-- 1 root root 27892 Jul 25 10:06 Manifest
-rw-r--r-- 1 portage portage 4845 Mar 23 21:37 bash-3.1.p17.ebuild
-rw-r--r-- 1 portage portage 5977 Mar 23 21:37 bash-3.2.p39.ebuild
-rw-r--r-- 1 portage portage 6151 Apr 5 14:37 bash-3.2.p48-r1.ebuild
-rw-r--r-- 1 portage portage 5988 Mar 23 21:37 bash-3.2.p48.ebuild
-rw-r--r-- 1 portage portage 5643 Apr 5 14:37 bash-4.0.p10-r1.ebuild
-rw-r--r-- 1 portage portage 6230 Apr 5 14:37 bash-4.0.p10.ebuild
-rw-r--r-- 1 portage portage 5648 Apr 14 05:52 bash-4.0.p17-r1.ebuild
-rw-r--r-- 1 portage portage 5532 Apr 8 10:21 bash-4.0.p17.ebuild
-rw-r--r-- 1 portage portage 5660 May 30 03:35 bash-4.0.p24.ebuild
-rw-r--r-- 1 root root 5660 Jul 25 09:43 bash-4.0.p28.ebuild
drwxr-xr-x 2 portage portage 2048 May 30 03:35 files
-rw-r--r-- 1 portage portage 468 Feb 9 04:35 metadata.xml
mars@marsmain /usr/portage/app-shells/bash $ cat metadata.xml
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE pkgmetadata SYSTEM "http://www.gentoo.org/dtd/metadata.dtd">
<pkgmetadata>
  <herd base-system>
    <use>
      <flag name='bashlogger'>Log ALL commands typed into bash; should ONLY be
        used in restricted environments such as honeypots</flag>
      <flag name='net'>Enable /dev/tcp/host/port redirection</flag>
      <flag name='plugins'>Add support for loading builtins at runtime via
        'enable'</flag>
    </use>
  </pkgmetadata>
mars@marsmain /usr/portage/app-shells/bash $ sudo /etc/init.d/bluetooth status
Passwoid:
* status: started
mars@marsmain /usr/portage/app-shells/bash $ ping -q -c1 en.wikipedia.org
PING rr.esams.wikimedia.org (91.198.174.2) 56(84) bytes of data.

--- rr.esams.wikimedia.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 2ms
rtt min/avg/max/mdev = 49.820/49.820/49.820/0.000 ms
mars@marsmain /usr/portage/app-shells/bash $ grep -i /dev/sda /etc/fstab | cut --fields=3
/dev/sda1 /boot
/dev/sda2 none
/dev/sda3 /
mars@marsmain /usr/portage/app-shells/bash $ date
Sat Aug 8 02:42:24 MSD 2009
mars@marsmain /usr/portage/app-shells/bash $ lsmod
Module Size Used by
rndis_wlan 23424 0
rndis_host 8696 1 rndis_wlan
cdc_ether 5672 1 rndis_host
usbnet 18688 3 rndis_wlan,rndis_host,cdc_ether
parport_pc 38424 0
lprlr 2388128 20
parport 39648 1 parport_pc
ltd0_wdt 12272 0
l2c_i801 9380 0
mars@marsmain /usr/portage/app-shells/bash $
```



System Calls

Una **system call** (llamada al sistema) es un punto de **entrada controlado al kernel**, permitiendo a un proceso solicitar que el kernel realice alguna operación en su nombre [KER](cap. 3). El kernel expone una gran cantidad de servicios accesibles por un programa vía el application programming interface (API) de system calls.



System Calls

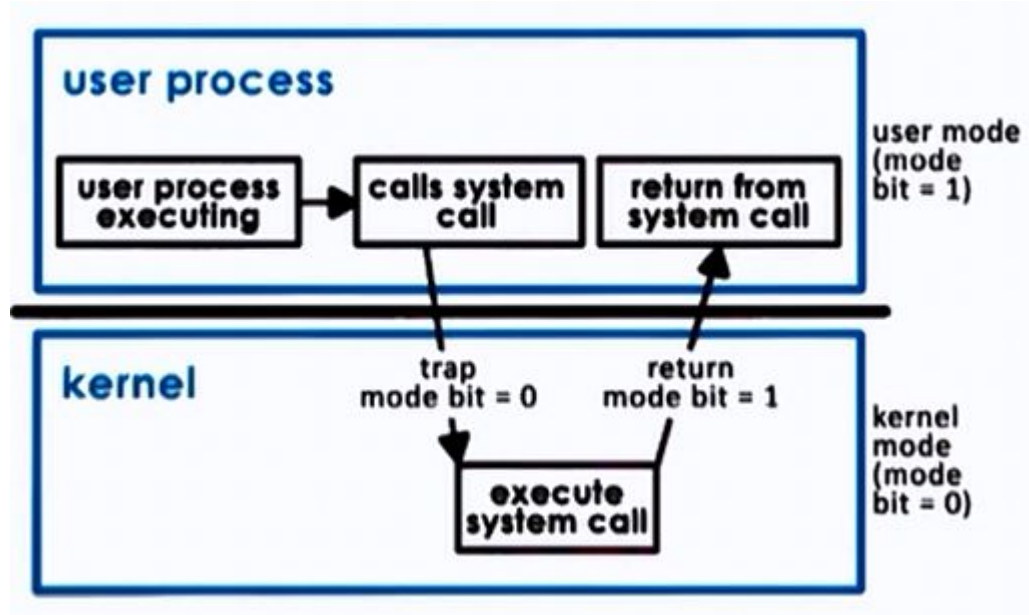
Algunas características generales de las system calls son:

Una system call cambia el modo del procesador de user mode a kernel mode, por ende la CPU podrá acceder al área protegida del kernel.

El conjunto de system calls es fijo. Cada system call está identificada por un único número, que por supuesto no es visible al programa, este sólo conoce su nombre.

Cada system call debe tener un conjunto de parámetros que especifican información que debe ser transferida desde el user space al kernel space.

System Calls



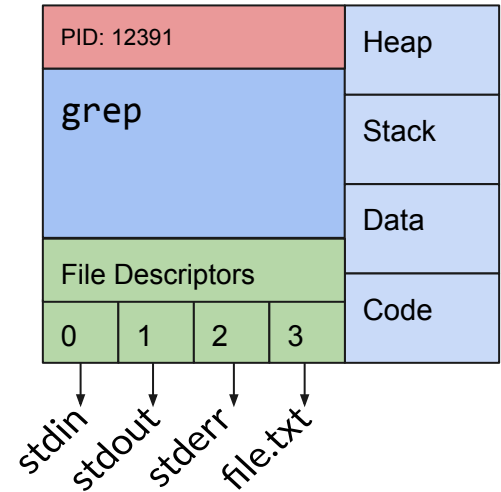
Procesos

Un proceso es un programa en ejecución

Es algo dinámico

Tienen una interna estructura propia

Todos los procesos menos el kernel viven en user-land





Procesos

En linux los procesos tiene un ciclo de vida típico

Dependen de 3 syscalls:

- `fork()`
- `wait()`
- `exec()`