

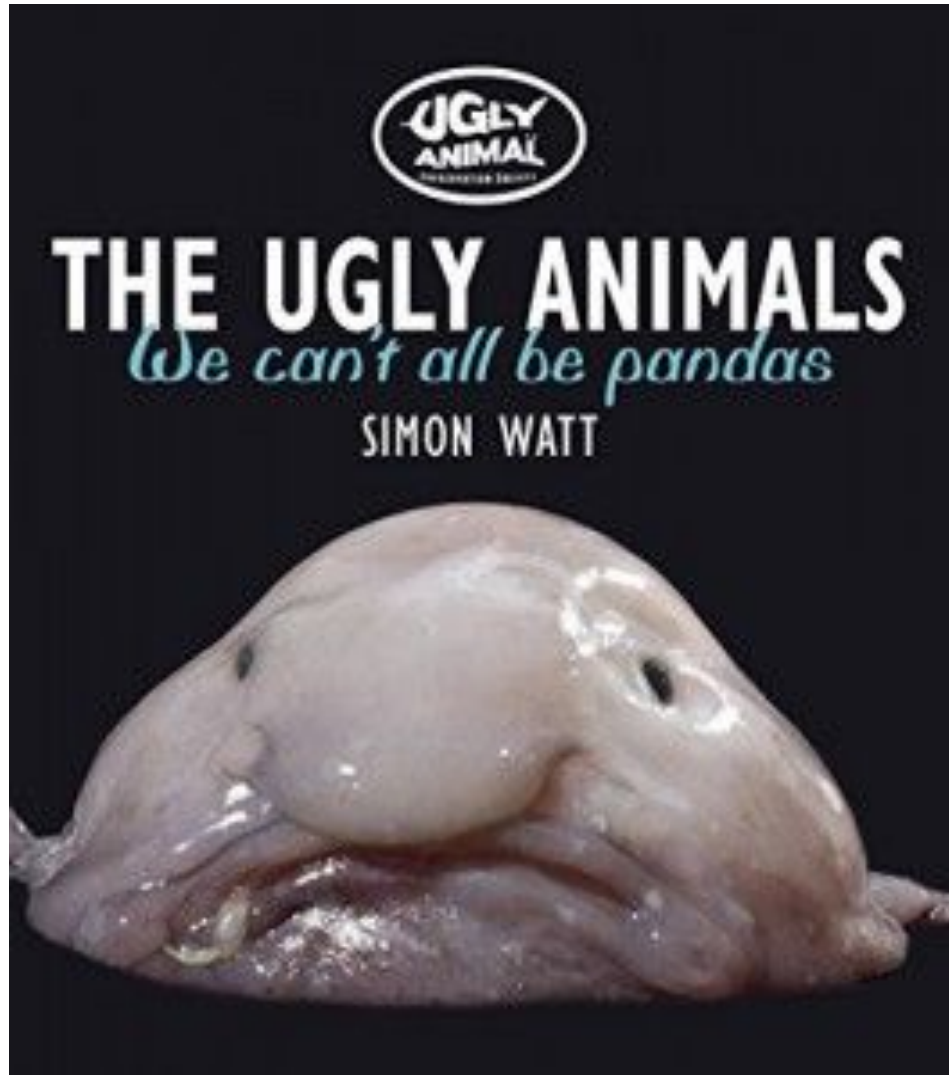


System calls

Blobfish



Blobfish: Mr Blobby



Blobfish at the sea bottom



Don't Be a Blobfish!

TOP DEFINITION



blobfish

1) an endangered **jelly** like creature at the bottom of the sea that is known for it's seemingly sad and depressed **facial** expression.

2) an individual who resides in an incompetent state of being

3) an individual who comes across as **mean**, angry, and/or unresponsive

2) *What a blobfish! He's **always shut up** in his **room** being unproductive.*

3) *You are not responding to my texts or calls. Why are you being such a blobfish to me?*

#mean #grumpy #depressed #sad #incompetent #unproductive #unresponsive #n
ice #happy #competent #productive #responsive

Be With a Blobfish!

Hashtag Collectibles®

PRODUCTS

SUBSCRIBE

FAQ

CONTACT

CART



\$ 649.00

~~\$ 749.50~~

BATH TIME BLOBBY

- Enough Blobfish to fill a bathtub!
- The blobbiest surprise surprise you can get this Valentine's.
- Includes 8 large Blobfish / 50 mini Blobfish.
 - Large: 12" X 20" each.
 - Mini: 8" X 5.5" each.

A very limited edition - get yours now while supplies last!

SIZE

Mini



QUANTITY

<https://www.hashtagcollectibles.com/collections/blobfish/products/bath-time-blobby>

Be With a Blobfish!



Hashtag Collectibles®

PRODUCTS

SUBSCRIBE

FAQ

CONTACT

CART



\$ 29.99

BLOBFISH SLIPPERS

They're adorable, they're fuzzy, and most importantly, they'll keep your toes warm. We are proud to introduce: the official Bloppy Slippers!

- One size fits all - adults

More available soon,
Sign up to be the first to know:

NOTIFY ME

SOLD OUT • \$ 29.99

<https://www.hashtagcollectibles.com/collections/blobfish/products/blobfish-slippers>

Typhlonus nasus: faceless fish



5000 m



System calls

Why?

Why?



compatibility

Why?

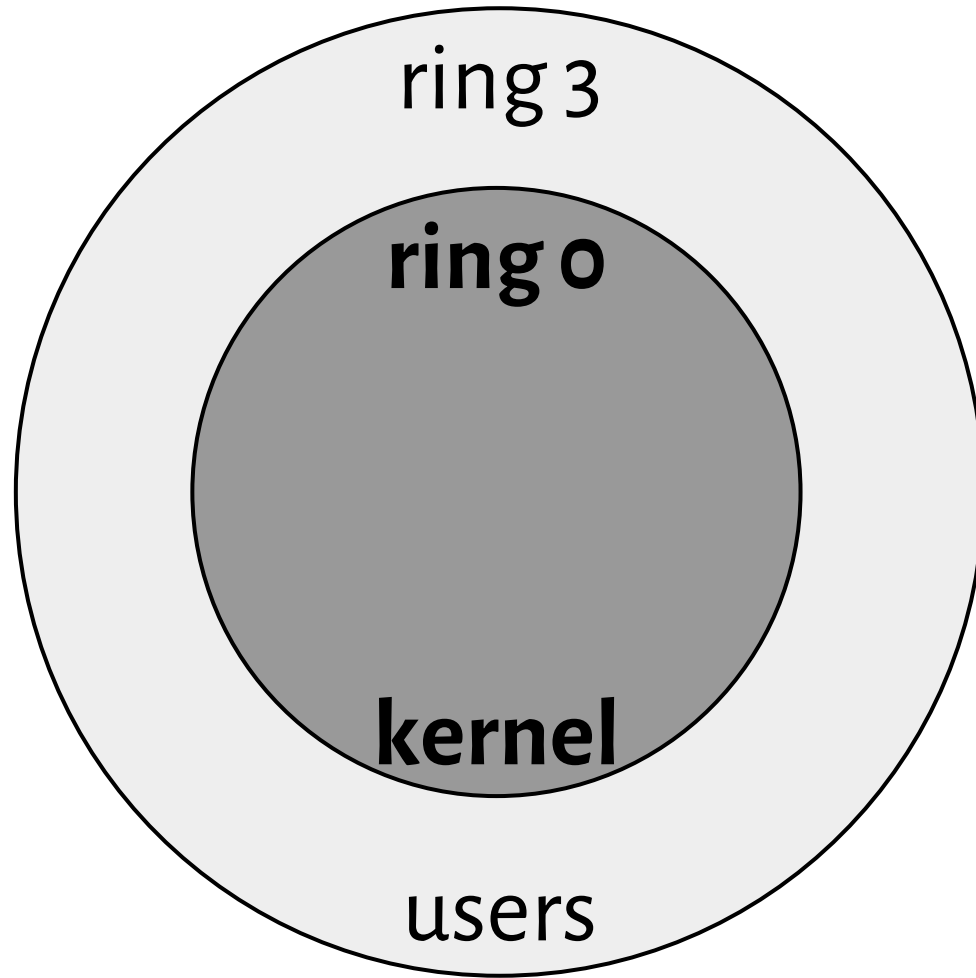


compatibility



users

Protection rings



What about rings 1 and 2?

User space

0x00007fffffffffff

0x0000000000000000



User Space

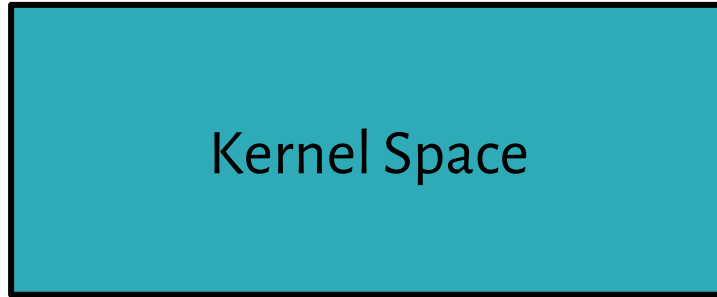


128 TiB

User space

0xffffffffffffffffffff

0xffff800000000000



128 TiB

0x00007fffffffffffff

0x0000000000000000



128 TiB

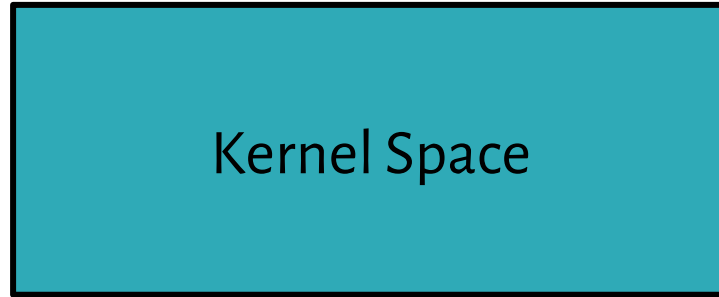
User space

0xffffffffffffffffffff

0xffff800000000000

0x00007fffffffffffff

0x0000000000000000



Kernel Space



User Space

128 TiB

> 16 000 000 TiB

128 TiB

What-and-why?

Operating systems offer processes running in *User Mode* **a set of interfaces** to interact with hardware devices such as the CPU, disks, printers, and so on.

System call

`brk()`

What-and-why?

Operating systems offer processes running in *User Mode* **a set of interfaces** to interact with hardware devices such as the CPU, disks, printers, and so on.

POSIX APIs (e.g. libc)

`malloc()`

`calloc()`


`free()`

System call

`brk()`

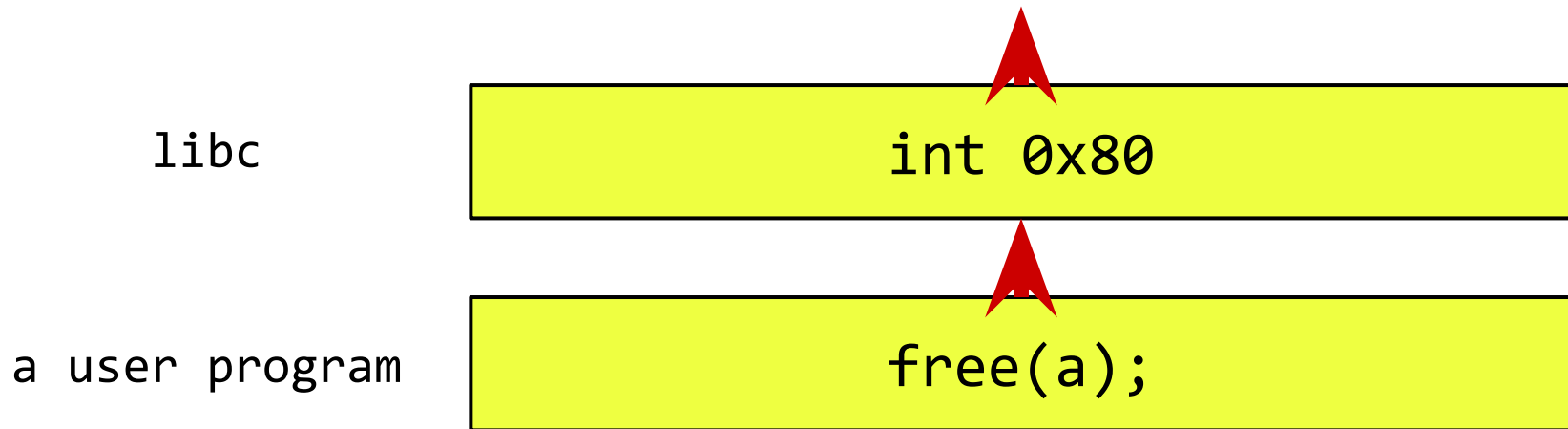
What-and-why?

a user program

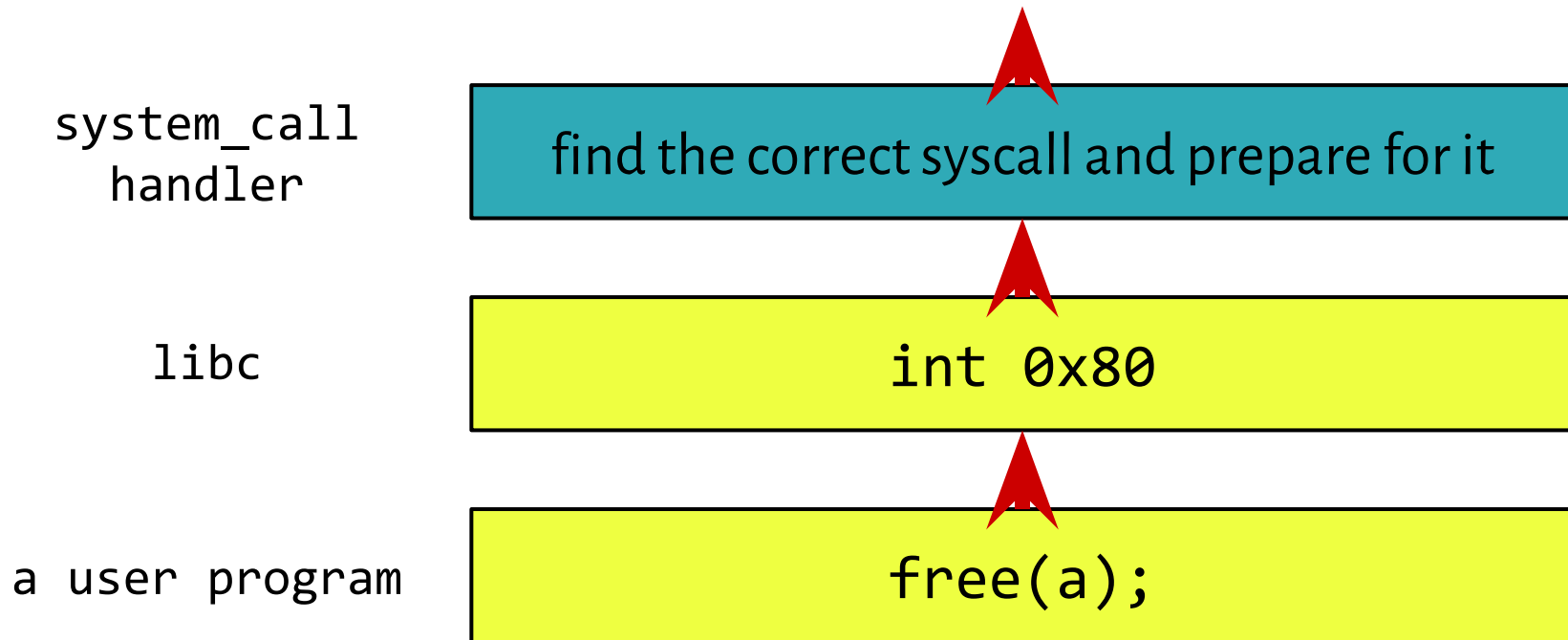


```
free(a);
```

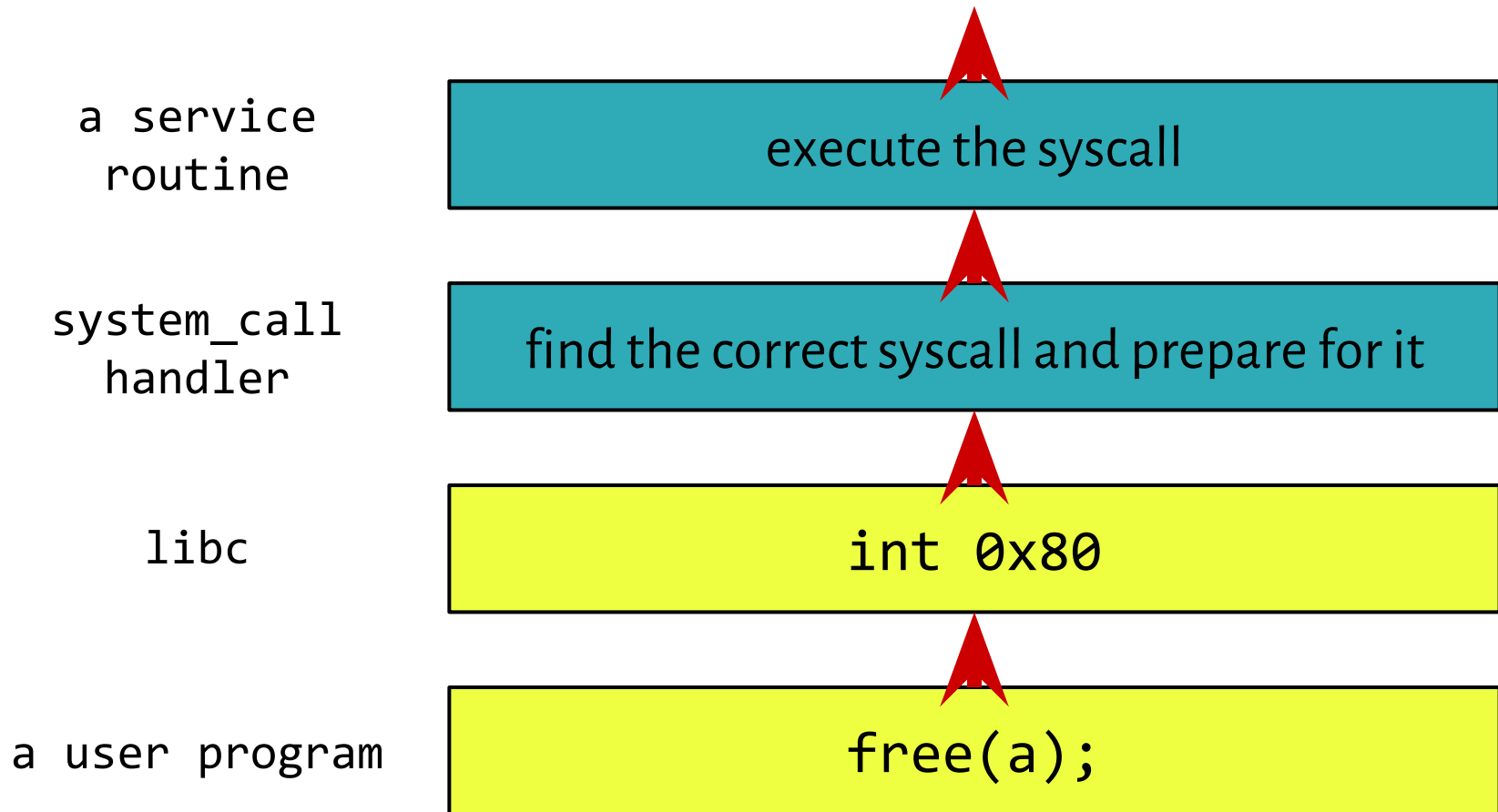

What-and-why?



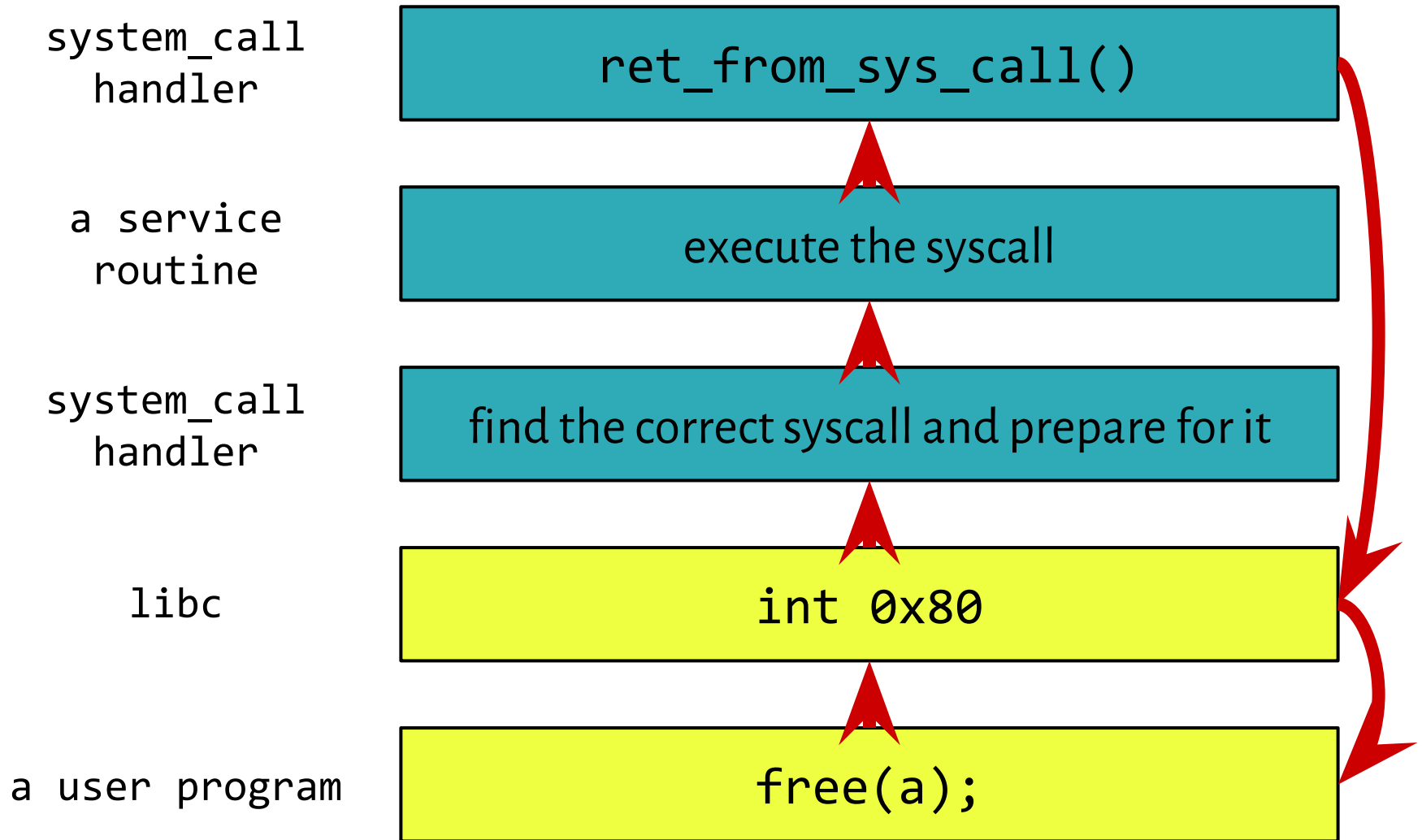
What-and-why?



What-and-why?



What-and-why?



Efficiency

The less switching, the better.

Results of a syscall are cached, e.g.: `getpid()`

<https://manybutfinite.com/post/system-calls/>

Operations are vectorized, e.g.: `readv()`

https://en.wikipedia.org/wiki/Vectored_I/O

Examples

Process Management

fork, waitpid, exit, ptrace...

Signals

sigaction, sigreturn, kill, alarm, pause...

File Management

create, open, seek, read, write, stat, rename...

File System Management

mkdir, link, unlink, mount, chdir, chroot...

Protection

chmod, setuid, chown, umask...

Time Management

time, stime, utime...

Assignment #3

Add a system call to the Process Manager server:

```
int pstree(pid_t pid, int uid)
```


Assignment #3

Add a system call to the Process Manager server:

```
int pstree(pid_t pid, int uid)
```

It should print a tree of the process <pid> that belongs to a user <uid>:

```
187
---188
-----191
---189
---190
```

where processes are presented in ascending order.

Assignment #3

- ★ Add a library function:

`/usr/src/lib/libc/misc/pstree.c`

- ★ Implement the syscall itself:

`/usr/src/minix/servers/pm/pstree.c`

- ★ Add a syscall number (and update the total number):

`/usr/src/minix/include/minix/callnr.h`

- ★ Modify the Makefile:

`/usr/src/minix/servers/pm/Makefile`

- ★ Add the declaration:

`/usr/src/minix/servers/pm/proto.h`

- ★ Bind implementation and declaration:

`usr/src/minix/servers/pm/table.c`

Assignment #3

/servers/pm/mproc.h

```
/* This table has one slot per process. It contains all the process management
 * information for each process. Among other things, it defines the text, data
 * and stack segments, uids and gids, and various flags. The kernel and file
 * systems have tables that are also indexed by process, with the contents
 * of corresponding slots referring to the same process in all three.
 */
#include <limits.h>
#include <minix/timers.h>
#include <signal.h>

#include <sys/cdefs.h>

/* Needs to be included here, for 'ps' etc */
#include "const.h"

EXTERN struct mproc {
    char mp_exitstatus; /* storage for status when process exits */
    char mp_sigstatus; /* storage for signal # for killed procs */
    pid_t mp_pid; /* process id */
    endpoint_t mp_endpoint; /* kernel endpoint id */
    pid_t mp_procgrp; /* pid of process group (used for signals) */
    pid_t mp_wpid; /* pid this process is waiting for */
    int mp_parent; /* index of parent process */
    int mp_tracer; /* index of tracer process, or NO_TRACER */
};
```

#include
"mproc.h"



struct mproc *ith_proc =
&mproc[i]

Assignment #3

```
if uid or pid is invalid then  
    it is not an error; just print nothing
```

Assignment #3

```
if uid or pid is invalid then  
    it is not an error; just print nothing
```

```
What uid should be taken into account:  
    effective or real?
```

Assignment #3

```
if uid or pid is invalid then  
    it is not an error; just print nothing
```

What uid should be taken into account:
effective or real?

In what situations should the program return -1?

Assignment #3

```
if uid or pid is invalid then  
    it is not an error; just print nothing
```

What uid should be taken into account:
effective or real?

In what situations should the program return -1?

How to sort the processes?

Assignment #3

```
if uid or pid is invalid then  
    it is not an error; just print nothing
```

What uid should be taken into account:
effective or real?

In what situations should the program return -1?

How to sort the processes?

How to get the user's group id?

Assignment #3

★ create your solution with the command:

```
$ diff -rupN <original>/usr/ <modified>/usr/ > ab123456.patch
```

Assignment #3

★ create your solution with the command:

```
$ diff -rupN <original>/usr/ <modified>/usr/ > ab123456.patch
```

★ apply your solution with the command:

```
# cd /; patch -p1 < ab123456.patch
```

Assignment #3

- ★ create your solution with the command:

```
$ diff -rupN <original>/usr/ <modified>/usr/ > ab123456.patch
```

- ★ apply your solution with the command:

```
# cd /; patch -p1 < ab123456.patch
```

- ★ the patch should be placed in the repository:
studenci/ab123456/zadanie3 by

7th May, 8 p.m.

