Linux overview 2

CS503: Operating systems, Spring 2019

Pedro Fonseca
Department of Computer Science
Purdue University

Admin (v2)

- Thursday (4/18):
  - OS + DS, system research
- Next week:
  - Tuesday (4/23): exam review
    - No quiz
  - Thursday (4/25): no class due to evening midterm
- Exam: **Thu 05/02, 8:00a - 10:00a @ FRNY B124**

What would you like to review?

- Memory management
  - Paging, segmentation
- Device management
- Clock and timer
- High-level IPC
- FS
- Virtualization: VMMs, Containers, TEEs
- Linux
- (You can be more specific)

Survey link
(also on Piazza)

Previous lecture

- History of Linux
- Scale
- Kernel structure
- Process management
- Debugging tools:
  - Debugger, strace, /proc/

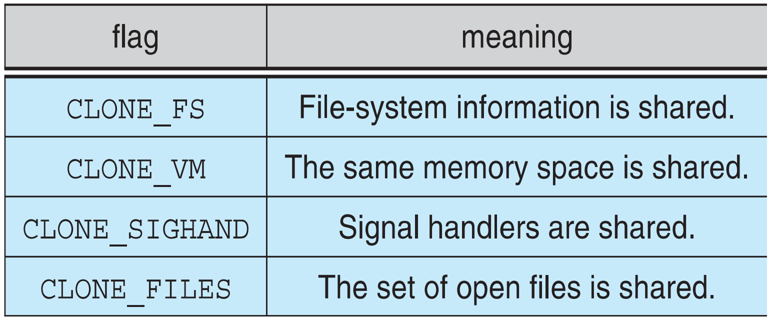
Clone system call

- int clone(int (*fn)(void *), void *child_stack,
  int flags, void *arg, …
  /* pid_t *ptid, struct user_desc *tls, pid_t *ctid */);

- “Unlike fork(2), clone() allows the child process to share parts of its execution context with the calling process, such as the memory space, the table of file descriptors, and the table of signal handlers.”

- “When the child process is created with clone(), it executes the function fn(arg). (This differs from fork(2), where execution continues in the child from the point of the fork(2) call.) The fn argument is a pointer to a function that is called by the child process at the beginning of its execution. The arg argument is passed to the fn function.”

- More info in “man clone”



flag	meaning
CLONE_FS	File-system information is shared.
CLONE_VM	The same memory space is shared.
CLONE_SIGHAND	Signal handlers are shared.
CLONE_FILES	The set of open files is shared.

Sysfs file system

- Usually mounted on /sys/
- Read and changes the kernel configuration
- Read system information
- Replaces the sysctl mechanism
- E.g.:
    - cat /sys/kernel/mm/hugepages/hugepages-1048576kB/free_hugepages

Memory management

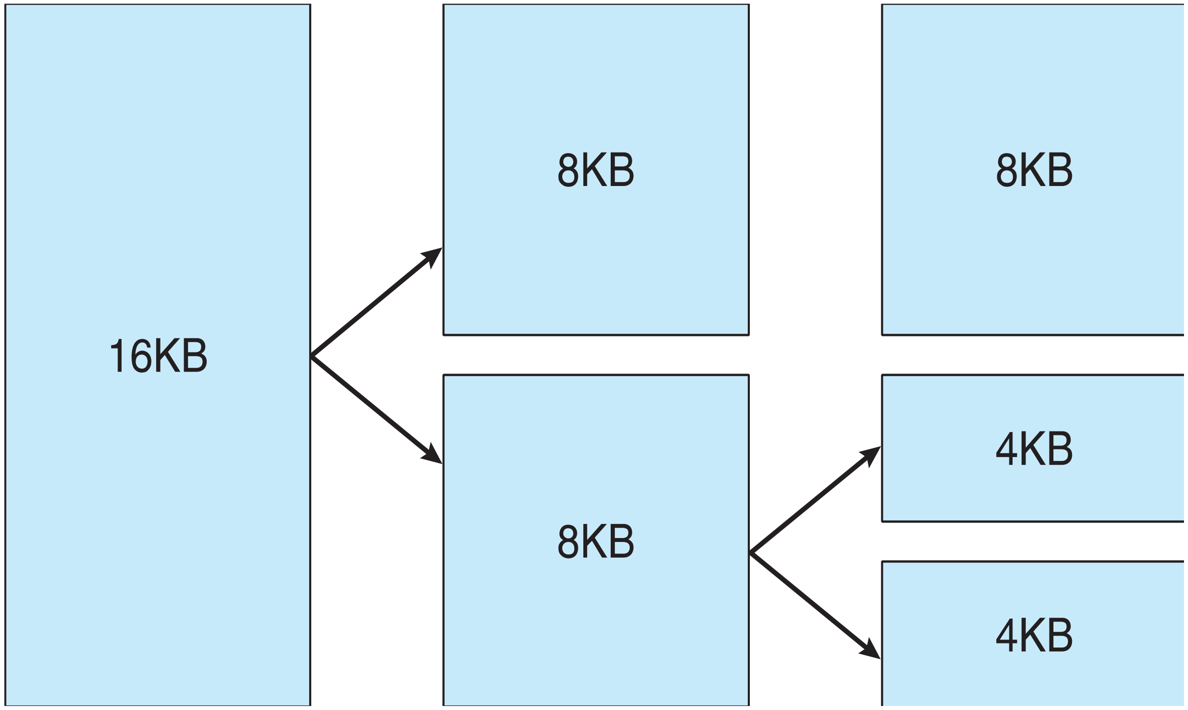
- Supports DMA
- Buddy allocator
- Slab allocator

Buddy heap

- Memory allocator that only allocates blocks of certain sizes:
  - Any block except the smallest can be divided into two smaller blocks of permitted sizes
  - Has one free list for each permitted size
- When the allocator receives a request for memory:
  - 1. It rounds the requested size up to a permitted size, and returns the first block from that size’s free list.
  - 2. If the free list for that size is empty, the allocator splits a block from a larger size and returns one of the pieces, adding the other to the appropriate free list.
- When blocks are recycled, there may be some attempt to merge adjacent blocks into ones of a larger permitted size (coalescence)
- The main advantage of the buddy system is that coalescence is cheap because the “buddy” of any free block can be calculated from its address

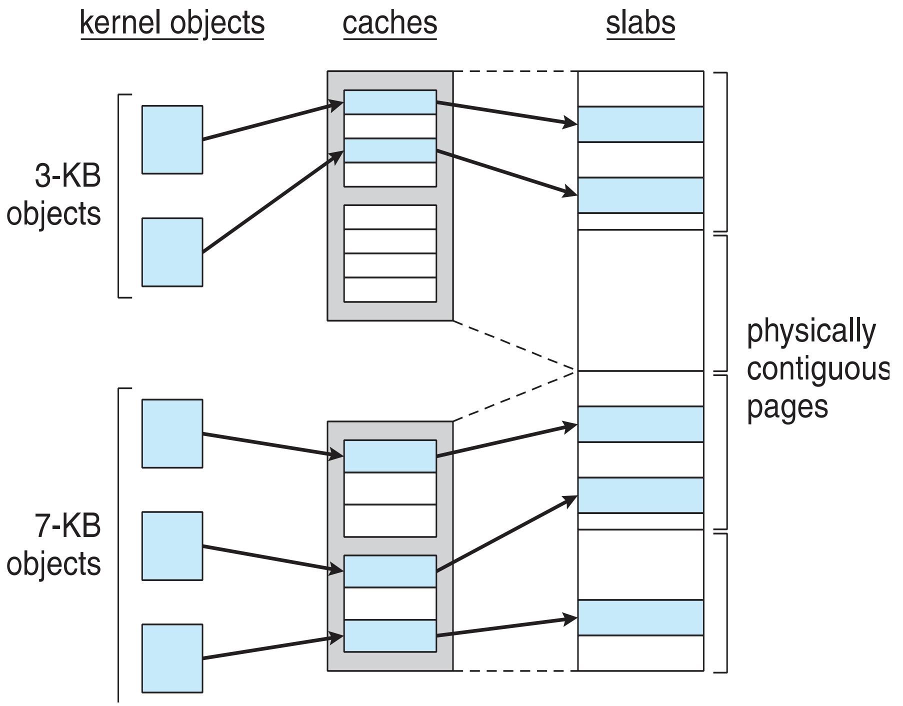
https://www.memorymanagement.org/mmref/alloc.html

Splitting of memory in a buddy heap



Slab allocator

- "Object factory"
- Slab allocator: set of caches of commonly used objects
  - Kept in an initialized state
  - Objects: in-memory inodes, semaphores, etc.
- Avoids spending time allocating, initializing, cleaning up, and freeing the same object
- The slab allocator aims to cache the freed object to reuse it later on



Discussion topics

- What is the importance of the block size in the buddy heap
- Other ways of dividing the block size in buddy heap?
- When is the slab allocator better?
- When is the buddy allocator better?

VM regions

- Backing store (where the pages came from)
  - Typically file backed or nothing (demand-zero memory)
- Regions reaction to writes:
  - Page sharing or copy-on-write

How to filter packets?

- How to filter packets efficiently?

Berkeley Packet Filter (BPF)

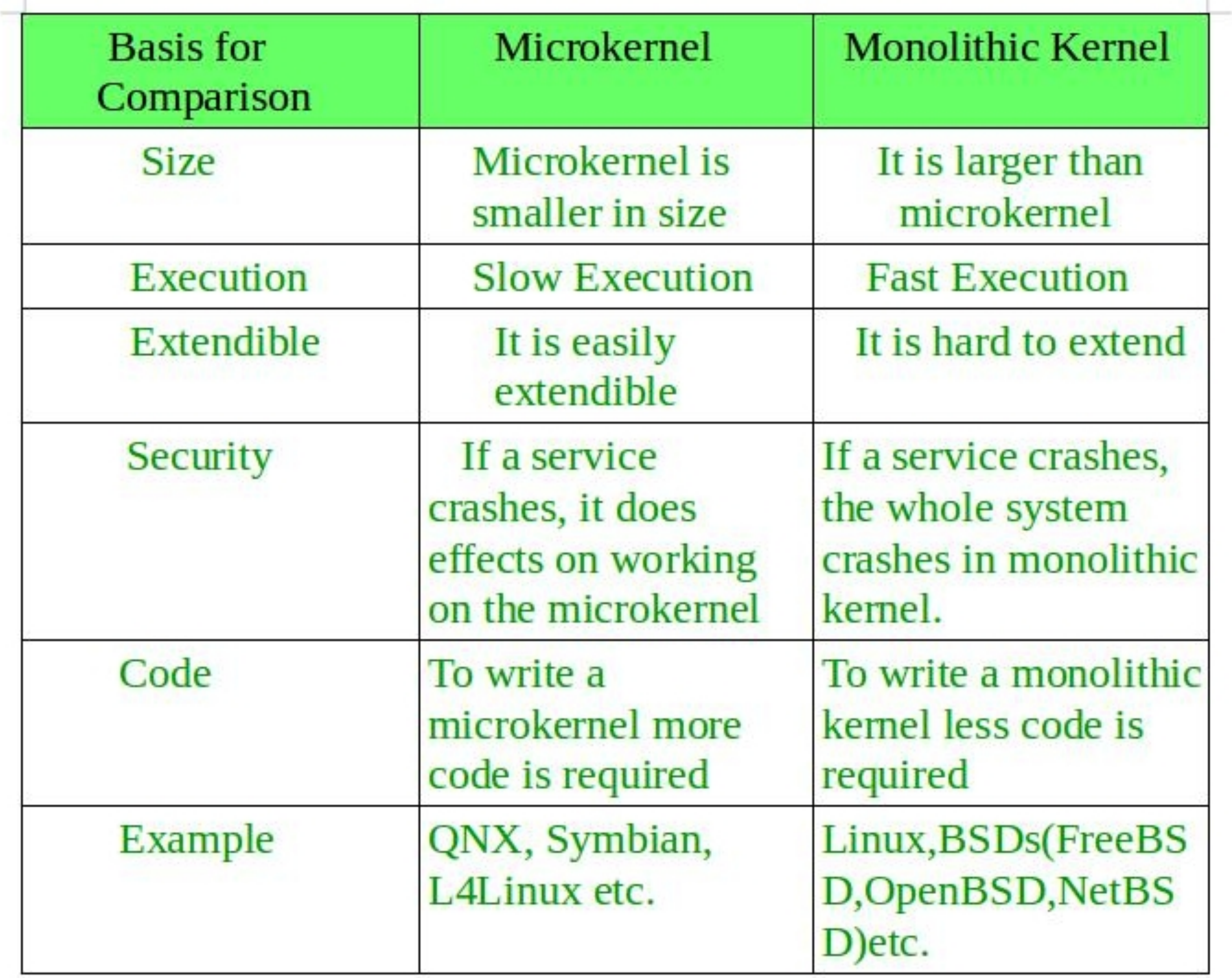
- Special-purpose virtual machine (register based filter evaluator) for filtering network packets
  - Now used in Linux for many other purposes
- Compiles a configuration of a filter into a program

Berkeley Packet Filter (BPF)

- Uses:
  - Traffic control
  - Sockets
  - Firewalling (xt_bpf module)
  - Tracing
  - Tracepoints
  - kprobe (dynamic tracing of a kernel function call)
  - cgroups
- Steven McCanne and Van Jacobson. 1993. The BSD packet filter: a new architecture for user-level packet capture. In Proceedings of the USENIX Winter 1993 Conference Proceedings on USENIX Winter 1993 Conference Proceedings (USENIX'93). USENIX Association, Berkeley, CA, USA, 2-2.

Kernel types

- Monolithic kernel
- Micro-kernels
- Q: what are the advantages of each?



Basis for Comparison	Microkernel	Monolithic Kernel
Size	Microkernel is smaller in size	It is larger than microkernel
Execution	Slow Execution	Fast Execution
Extendible	It is easily extendible	It is hard to extend
Security	If a service crashes, it does effects on working on the microkernel	If a service crashes, the whole system crashes in monolithic kernel.
Code	To write a microkernel more code is required	To write a monolithic kernel less code is required
Example	QNX, Symbian, L4Linux etc.	Linux,BSDs(FreeBSD,OpenBSD,NetBSD)etc.

Summary

- Clone system call
- Sys file system
- Virtual memory
- Buddy allocator
- Slab allocator
- BPF
- Micro-kernel vs. monolithic kernel