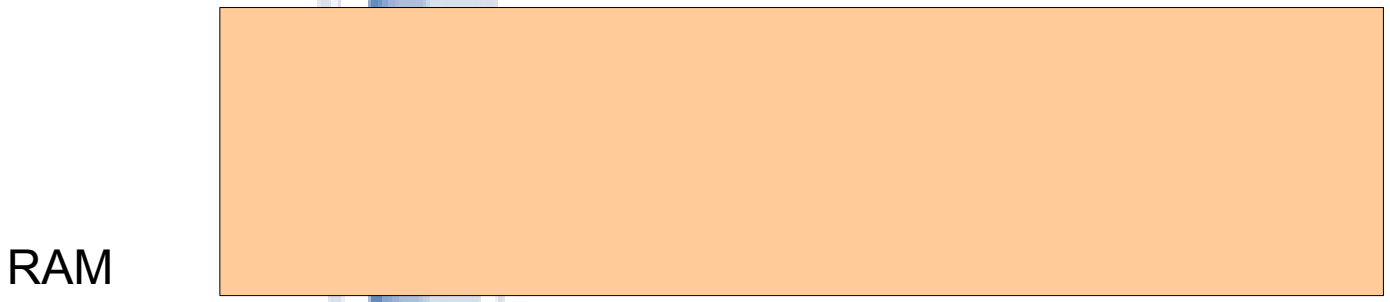
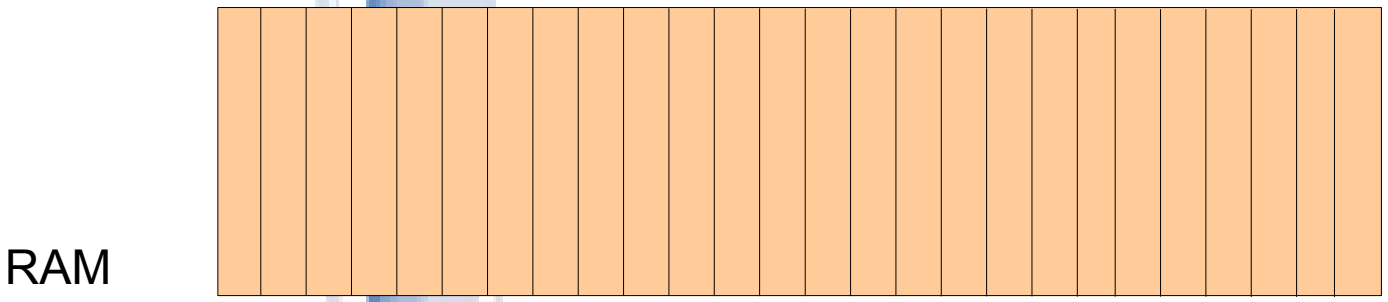
Virtual Memory

Physical Memory



- The kernel keeps track of how it is used

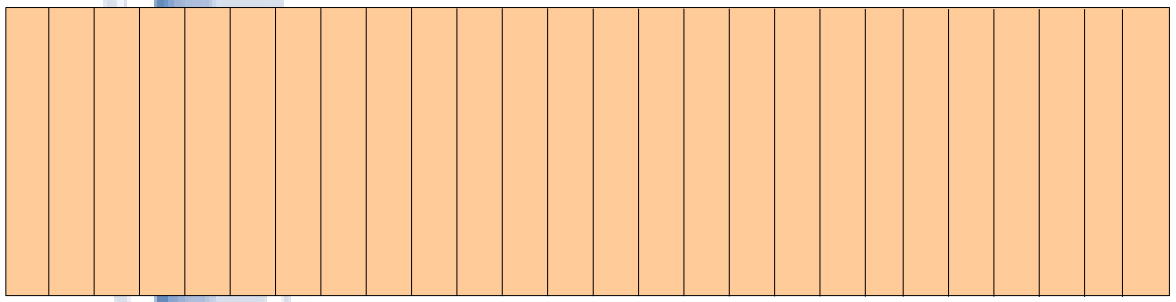
Physical Memory



Kernel and hardware typically with it in pages

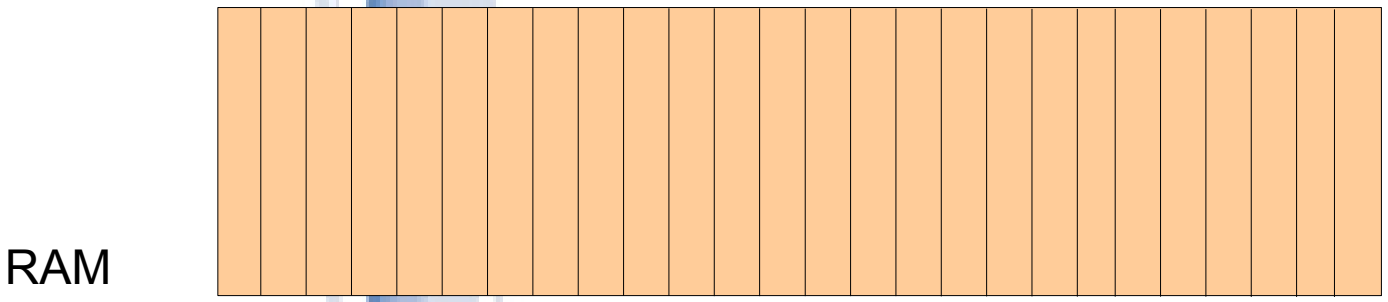
- Usually-
- 4k on 32-bit architectures
- 8k on 64-bit architectures

Physical Memory



On a machine with 4KB pages and 1GB RAM, there are 262,144 pages

Physical Memory

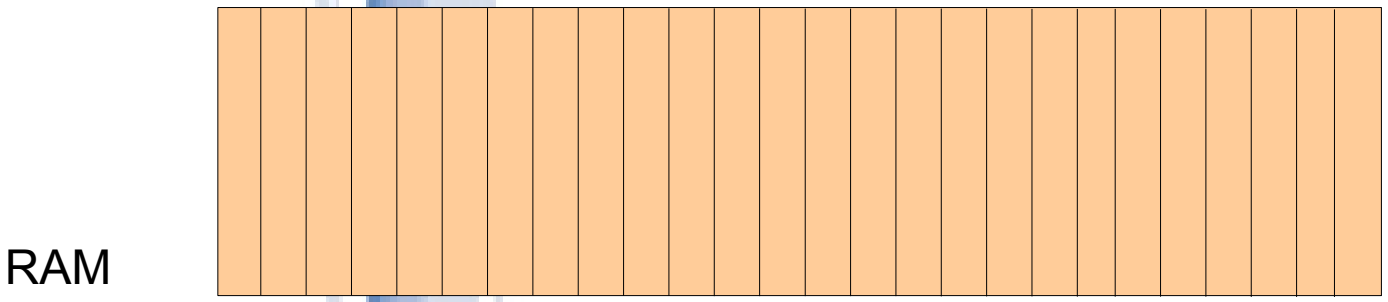


On a machine with 4KB pages and 1GB RAM, there are 262,144 pages

Every page is represented with a

`struct page`

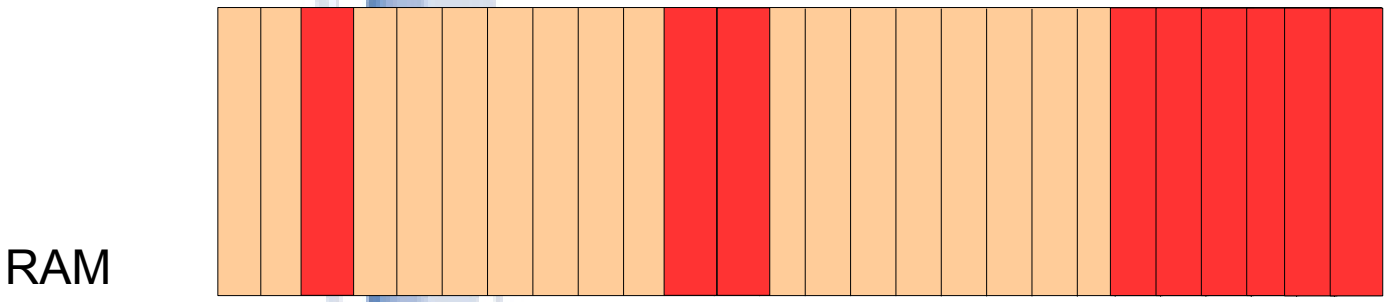
Physical Memory



Possible owners for each page include:

- User-space processes
- Allocated kernel data
- Static kernel code
- Page cache, etc

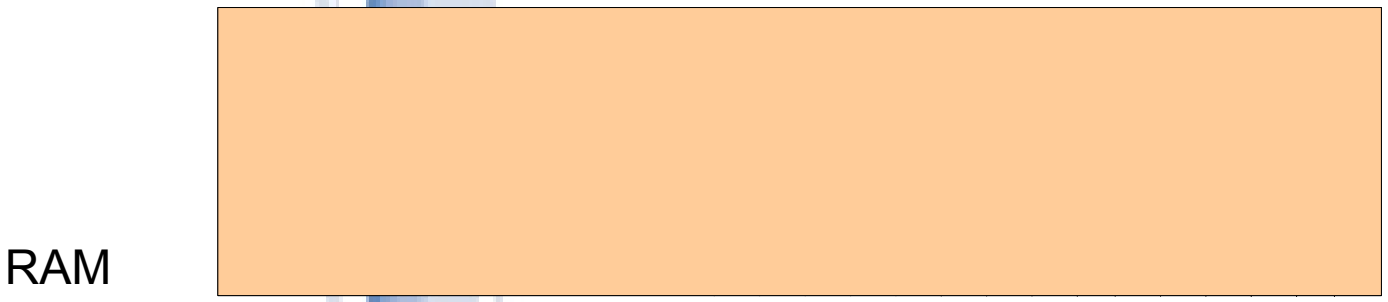
Physical Memory



Some pages get special treatment

- DMA capable (ZONE_DMA)
- Highmem (ZONE_HIGHMEM)
- Others are ZONE_NORMAL

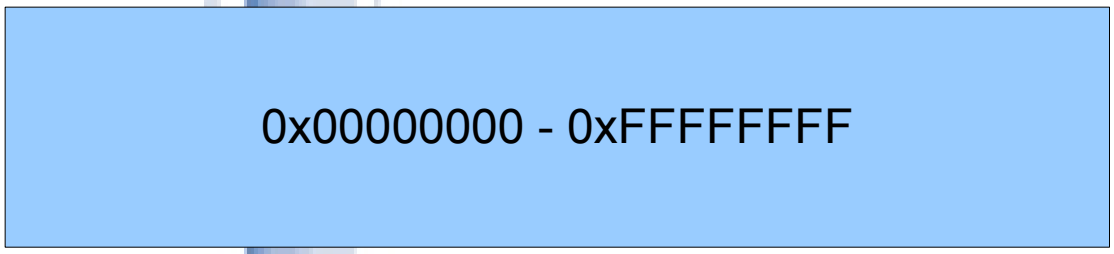
Physical Memory



- **User processes don't see any of this**
- They don't see physical memory at all
- They see a virtual address space all their own

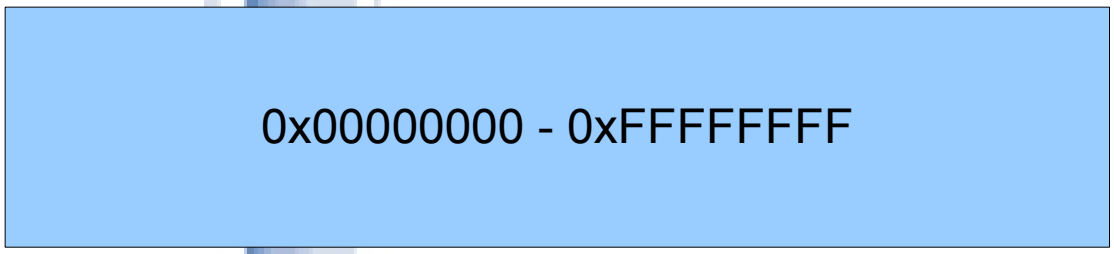
What is Virtual Memory?

- A technique that gives each process the illusion of a private, contiguous address space

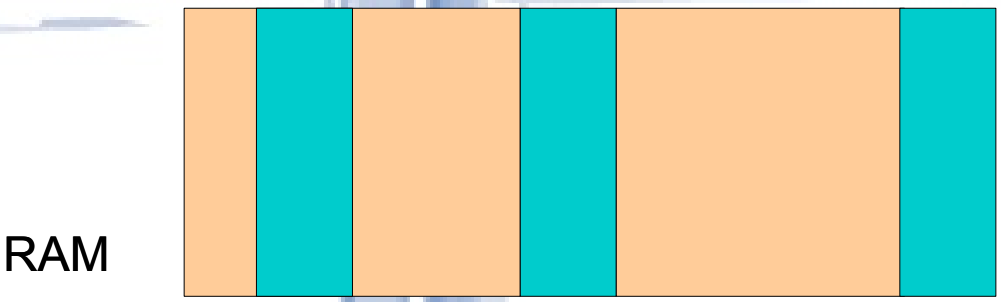


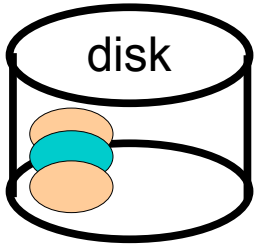
What is Virtual Memory?

- A technique that gives each process the illusion of a private, contiguous address space

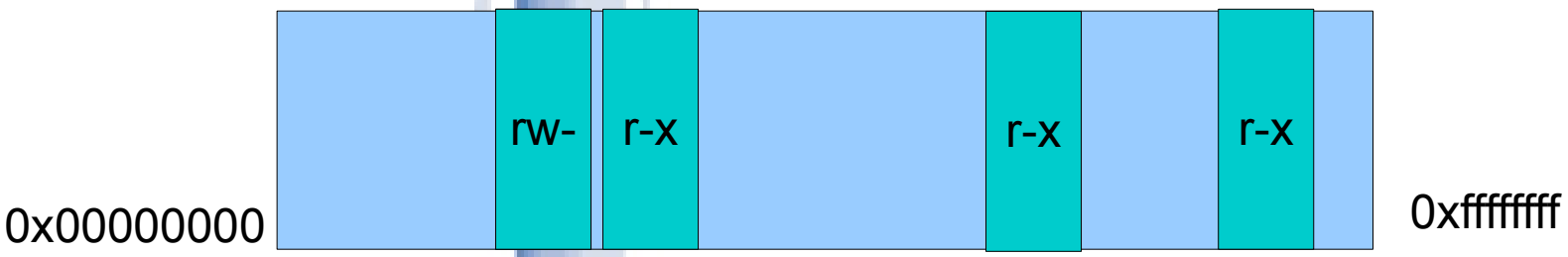


- From fragments of physical RAM and disk



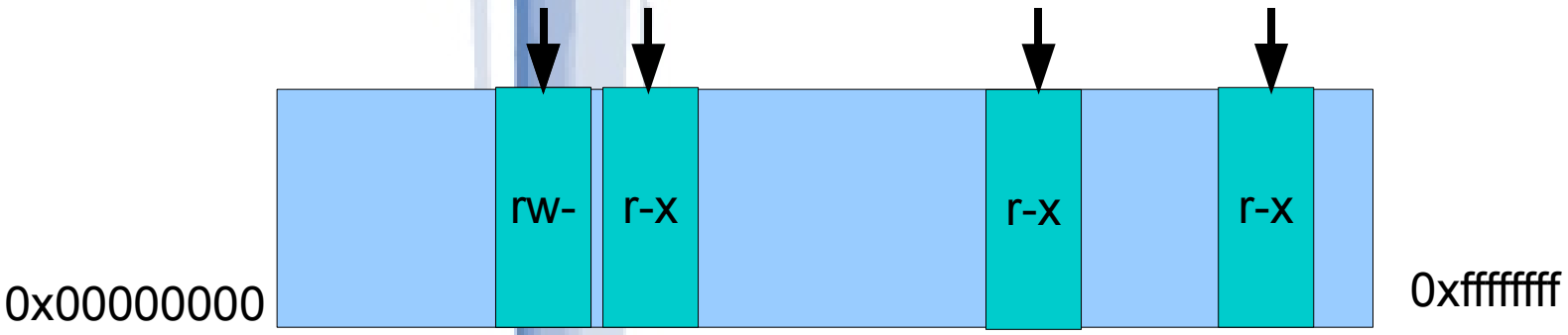


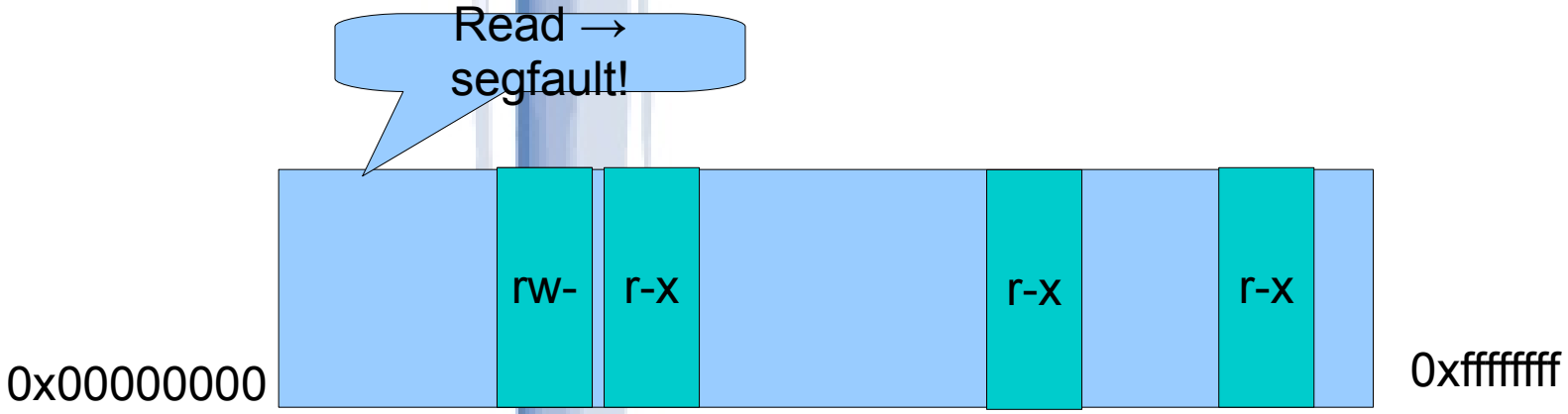
Even though the whole space is addressable,
only certain areas are legal for *r-w-x*

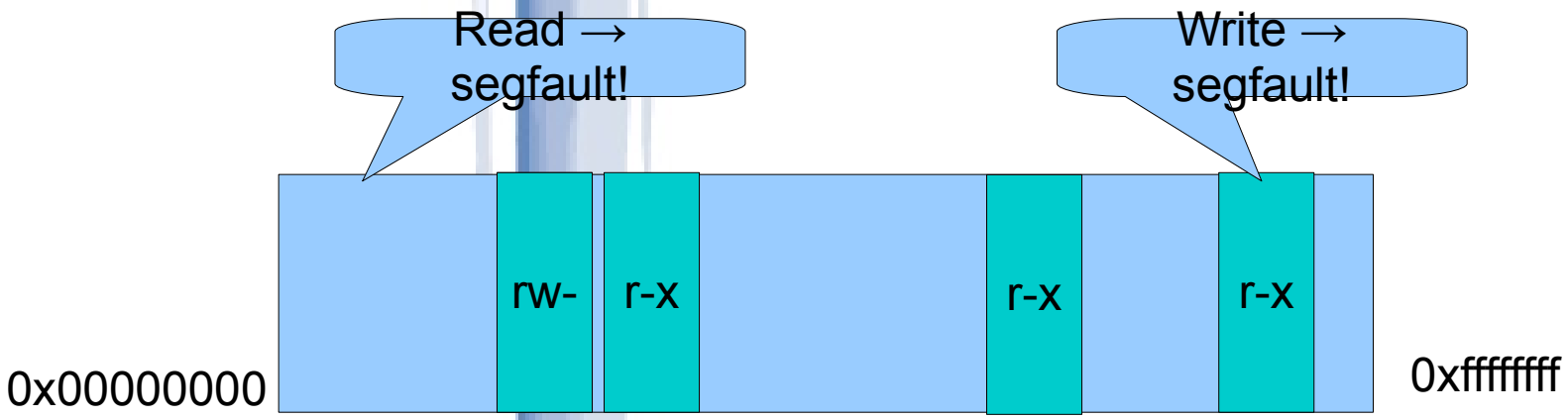


These legally addressable areas are called

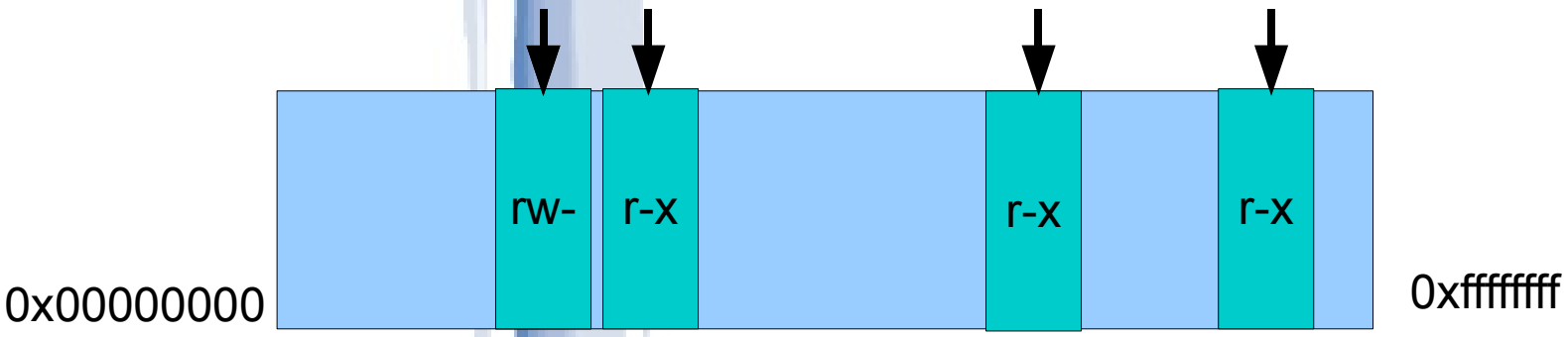
- ***Virtual Memory Areas***




Read →
segfault!
rw-
r-x
r-x
r-x
0x00000000
0xffffffff


Read → segfault!
Write → segfault!
rw-
r-x
r-x
r-x
0x00000000
0xffffffff

Virtual Memory Areas




```
[thanson@linux02]$ pmap 20580
0043e000     88K r-x--  /lib/ld-2.3.4.so
00454000      4K r-x--  /lib/ld-2.3.4.so
00455000      4K rwx--  /lib/ld-2.3.4.so
00458000   1176K r-x--  /lib/tls/libc-2.3.4.so
0057e000      8K r-x--  /lib/tls/libc-2.3.4.so
00580000      8K rwx--  /lib/tls/libc-2.3.4.so
00582000      8K rwx--    [ anon ]
08048000      4K r-x--  /home/thanson/trash/csleep
08049000      4K rw---  /home/thanson/trash/csleep
b7f9d000      4K rw---    [ anon ]
bffd1000    188K rw---    [ stack ]
ffffe000      4K -----    [ anon ]
 total     1500K
```

virtual → physical

- Software deals with **virtual** addresses
- Hardware needs **physical** addresses

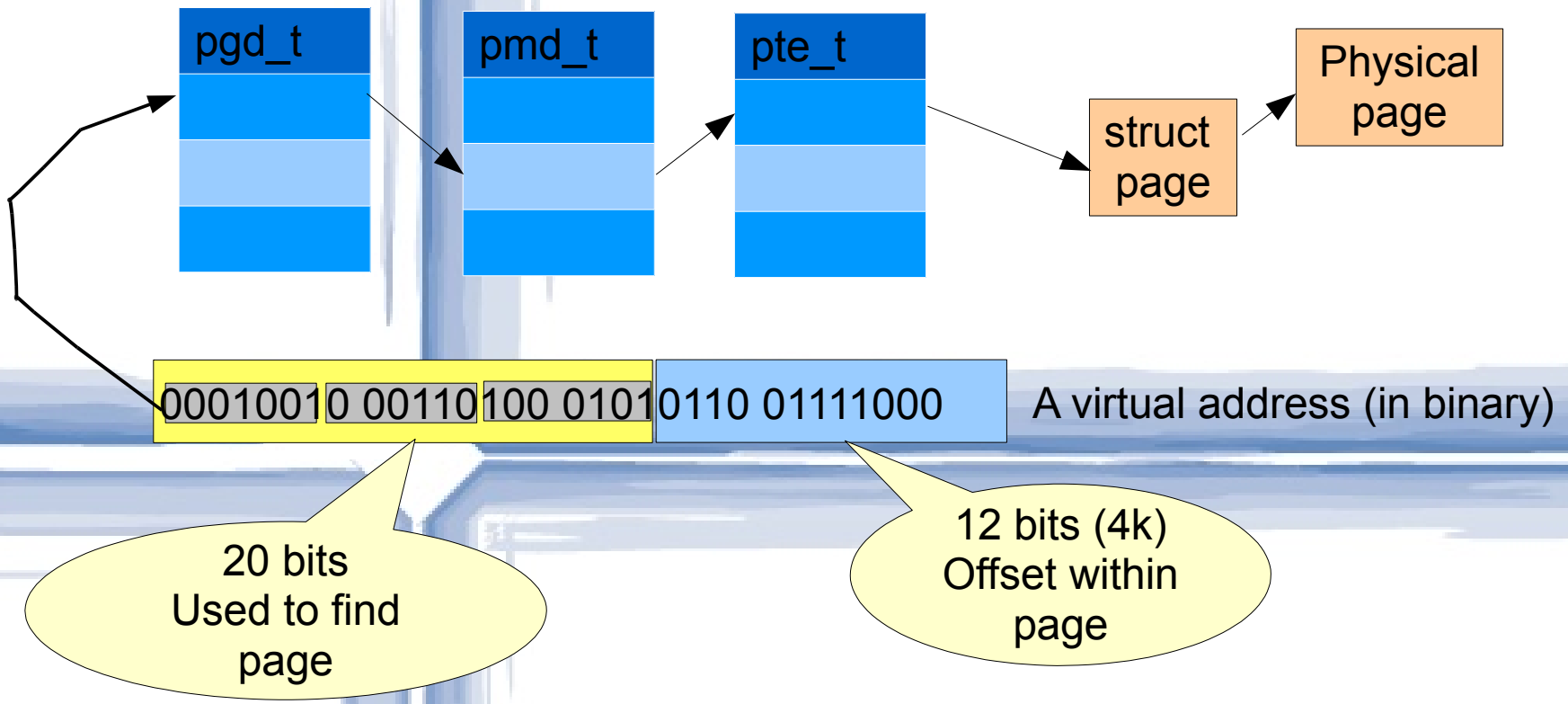
0x12345678 → ???

Page tables

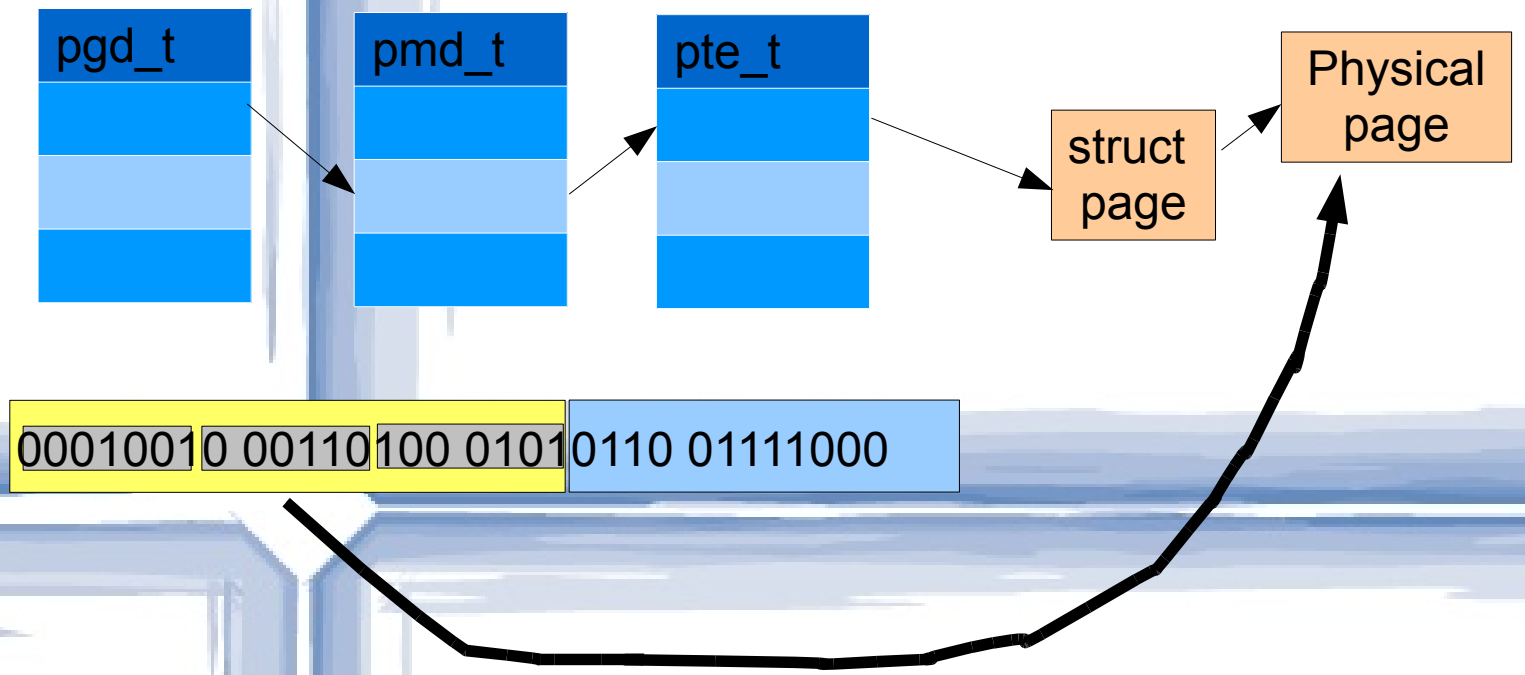
- Every process has a set of **page tables**

Page tables

- Linux uses a **3-level** page table scheme

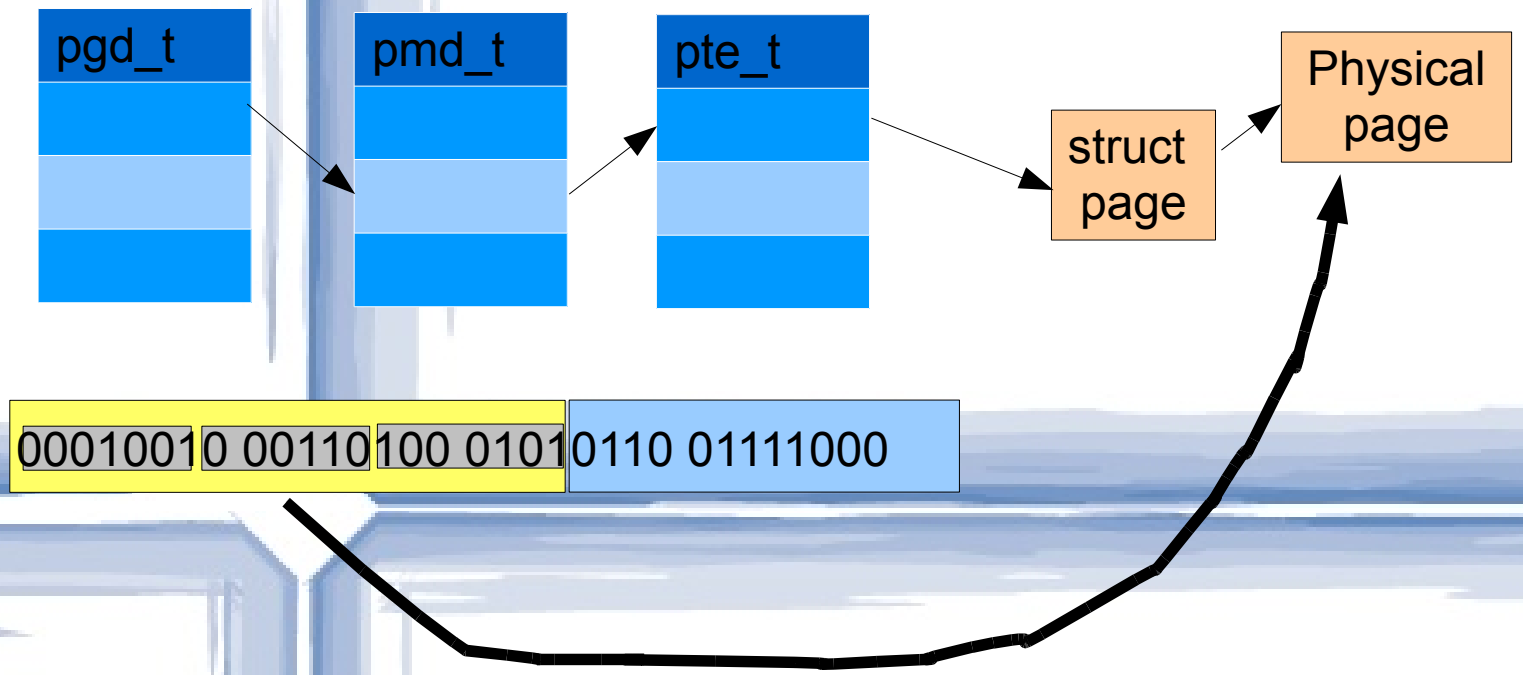


Page tables



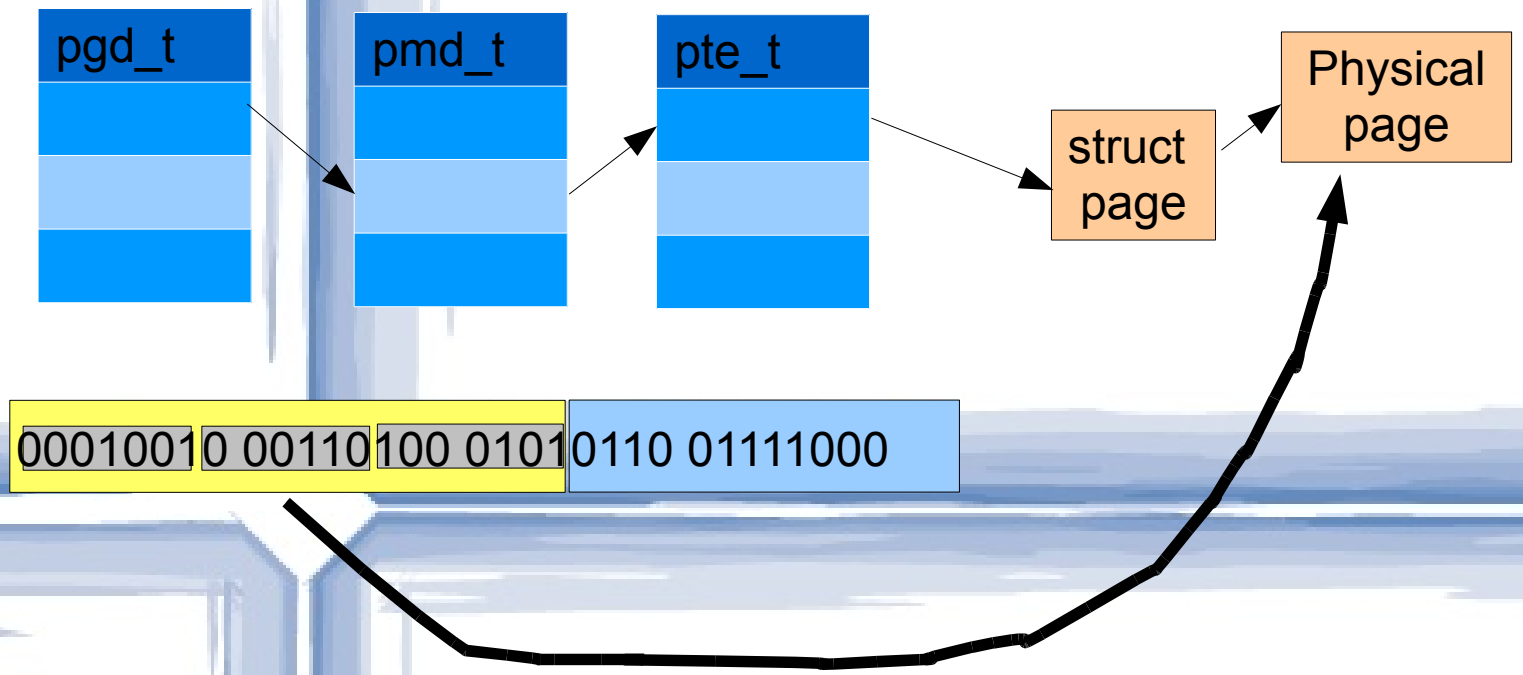
One lookup would be a lot faster!

Page tables



The hardware has a small cache called the **TLB**
Translation Lookaside Buffer

Page tables



Successive reads on the same page (or other recently used pages) will be TLB hits and avoid the costly page table lookups

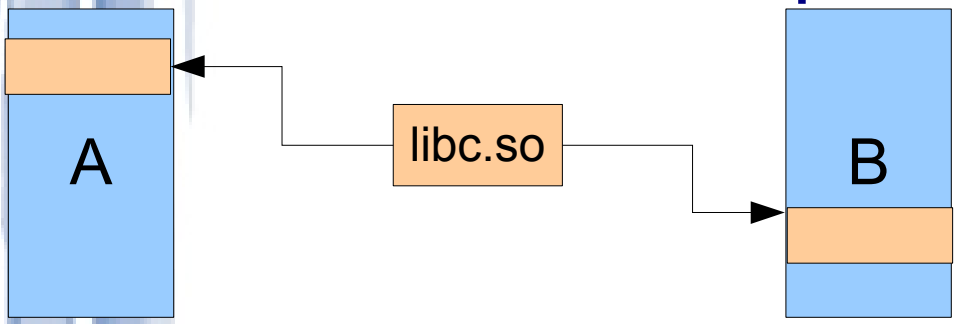
The page tables may indicate the page is not in RAM...

- it may be **swapped out**
  - MMU pagefaults into kernel to load it from disk
- it may be **“demand paged”**
  - it was allocated or mapped, but never touched yet
  - the kernel didn't bother allocating a page for it until first attempt to access it

How much memory does my process take?

- *Your process is a soup of pages.*
- Some pages on disk

- Some pages in RAM
- Some pages shared with other processes.



How much memory does my process take?

- **Virtual size (vsize) = all virtual memory areas**

regardless of whether they are in physical memory, or on disk, or shared with other processes

- **Resident set size (rss) = physical memory**

but one physical page shared by two processes counts in both

```
$ ps -ofname,pid,vsize,rss
COMMAND     PID   VSZ  RSS
bash       4027  5244 1440
```

How much memory does my process take?

```
$ pmap 4027
4027:   -bash
0043e000      88K r-x--  /lib/ld-2.3.4.so
00454000       4K r-x--  /lib/ld-2.3.4.so
00455000       4K rwx--  /lib/ld-2.3.4.so
00458000    1176K r-x--  /lib/tls/libc-2.3.4.so
0057e000       8K r-x--  /lib/tls/libc-2.3.4.so
00580000       8K rwx--  /lib/tls/libc-2.3.4.so
00582000       8K rwx--    [ anon ]
005ab000       8K r-x--  /lib/libdl-2.3.4.so
005ad000       4K r-x--  /lib/libdl-2.3.4.so
005ae000       4K rwx--  /lib/libdl-2.3.4.so
006b4000      12K r-x--  /lib/libtermcap.so.2.0.8
006b7000       4K rwx--  /lib/libtermcap.so.2.0.8
00d00000      36K r-x--  /lib/libnss_files-2.3.4.so
00d09000       4K r-x--  /lib/libnss_files-2.3.4.so
00d0a000       4K rwx--  /lib/libnss_files-2.3.4.so
08047000     580K r-x--  /bin/bash
080d8000      24K rw---  /bin/bash
080de000      20K rw---    [ anon ]
09baa000     264K rw---    [ anon ]
b7d1f000       8K rw---    [ anon ]
b7d21000      24K r--s-  /usr/lib/gconv/gconv-modules.cache
b7d27000    2048K r----  /usr/lib/locale/locale-archive
b7f27000       8K rw---    [ anon ]
bff20000     896K rw---    [ stack ]
ffffe000       4K -----    [ anon ]
 total      5248K
```

```
$ ps -ofname,pid,vsize,rss
COMMAND     PID   VSZ  RSS
bash       4027  5244 1440
```

How much memory does my process take?

```
$ pmap 4027
4027:   -bash
0043e000      88K r-x--  /lib/ld-2.3.4.so
00454000       4K r-x--  /lib/ld-2.3.4.so
00455000       4K rwx--  /lib/ld-2.3.4.so
00458000    1176K r-x--  /lib/tls/libc-2.3.4.so
0057e000       8K r-x--  /lib/tls/libc-2.3.4.so
00580000       8K rwx--  /lib/tls/libc-2.3.4.so
00582000       8K rwx--    [ anon ]
005ab000       8K r-x--  /lib/libdl-2.3.4.so
005ad000       4K r-x--  /lib/libdl-2.3.4.so
005ae000       4K rwx--  /lib/libdl-2.3.4.so
006b4000      12K r-x--  /lib/libtermcap.so.2.0.8
006b7000       4K rwx--  /lib/libtermcap.so.2.0.8
00d00000      36K r-x--  /lib/libnss_files-2.3.4.so
00d09000       4K r-x--  /lib/libnss_files-2.3.4.so
00d0a000       4K rwx--  /lib/libnss_files-2.3.4.so
08047000     580K r-x--  /bin/bash
080d8000      24K rw---  /bin/bash
080de000      20K rw---    [ anon ]
09baa000     264K rw---    [ anon ]
b7d1f000       8K rw---    [ anon ]
b7d21000      24K r--s-  /usr/lib/gconv/gconv-modules.cache
b7d27000    2048K r----  /usr/lib/locale/locale-archive
b7f27000       8K rw---    [ anon ]
bff20000     896K rw---    [ stack ]
ffffe000       4K -----    [ anon ]
 total      5248K
```

Virtual Memory Areas (VMA's)
total size adds up to the VSZ

```
$ ps -ofname,pid,vsize,rss
COMMAND     PID    VSZ   RSS
bash       4027   5244  1440
```

How much memory does my process take?

```
$ pmap 4027
4027:   -bash
0043e000     88K r-x--  /lib/ld-2.3.4.so
00454000      4K r-x--  /lib/ld-2.3.4.so
00455000      4K rwx--  /lib/ld-2.3.4.so
00458000   1176K r-x--  /lib/tls/libc-2.3.4.so
0057e000      8K r-x--  /lib/tls/libc-2.3.4.so
00580000      8K rwx--  /lib/tls/libc-2.3.4.so
00582000      8K rwx--    [ anon ]
005ab000      8K r-x--  /lib/libdl-2.3.4.so
005ad000      4K r-x--  /lib/libdl-2.3.4.so
005ae000      4K rwx--  /lib/libdl-2.3.4.so
006b4000     12K r-x--  /lib/libtermcap.so.2.0.8
006b7000      4K rwx--  /lib/libtermcap.so.2.0.8
00d00000     36K r-x--  /lib/libnss_files-2.3.4.so
00d09000      4K r-x--  /lib/libnss_files-2.3.4.so
00d0a000      4K rwx--  /lib/libnss_files-2.3.4.so
08047000    580K r-x--  /bin/bash
080d8000     24K rw---  /bin/bash
080de000     20K rw---    [ anon ]
09baa000    264K rw---    [ anon ]
b7d1f000      8K rw---    [ anon ]
b7d21000     24K r--s-  /usr/lib/gconv/gconv-modules
b7d27000   2048K r----  /usr/lib/locale/locale-archive
b7f27000      8K rw---    [ anon ]
bff20000    896K rw---    [ stack ]
ffffe000      4K -----    [ anon ]
 total     5248K
```

```
$ pmap 3998
3998:   -bash
0043e000     88K r-x--  /lib/ld-2.3.4.so
00454000      4K r-x--  /lib/ld-2.3.4.so
00455000      4K rwx--  /lib/ld-2.3.4.so
00458000   1176K r-x--  /lib/tls/libc-2.3.4.so
0057e000      8K r-x--  /lib/tls/libc-2.3.4.so
00580000      8K rwx--  /lib/tls/libc-2.3.4.so
00582000      8K rwx--    [ anon ]
005a0000     36K r-x--  /lib/libnss_files-2.3.4.so
005a9000      4K r-x--  /lib/libnss_files-2.3.4.so
005aa000      4K rwx--  /lib/libnss_files-2.3.4.so
005ab000      8K r-x--  /lib/libdl-2.3.4.so
005ad000      4K r-x--  /lib/libdl-2.3.4.so
005ae000      4K rwx--  /lib/libdl-2.3.4.so
006b4000     12K r-x--  /lib/libtermcap.so.2.0.8
006b7000      4K rwx--  /lib/libtermcap.so.2.0.8
08047000    580K r-x--  /bin/bash
080d8000     24K rw---  /bin/bash
080de000     20K rw---    [ anon ]
08fae000    264K rw---    [ anon ]
b7d1b000      8K rw---    [ anon ]
b7d1d000     24K r--s-  /usr/lib/gconv/gconv-modules.cache
b7d23000   2048K r----  /usr/lib/locale/locale-archive
b7f23000      8K rw---    [ anon ]
bfeb0000   1344K rw---    [ stack ]
ffffe000      4K -----    [ anon ]
 total     5696K
```

Two bash processes running at the same time

How much memory does my process take?

```
$ pmap 4027
4027:   -bash
0043e000     88K r-x--  /lib/ld-2.3.4.so
00454000      4K r-x--  /lib/ld-2.3.4.so
00455000      4K rwx--  /lib/ld-2.3.4.so
00458000   1176K r-x--  /lib/tls/libc-2.3.4.so
0057e000      8K r-x--  /lib/tls/libc-2.3.4.so
00580000      8K rwx--  /lib/tls/libc-2.3.4.so
00582000      8K rwx--    [ anon ]
005ab000      8K r-x--  /lib/libdl-2.3.4.so
005ad000      4K r-x--  /lib/libdl-2.3.4.so
005ae000      4K rwx--  /lib/libdl-2.3.4.so
006b4000     12K r-x--  /lib/libtermcap.so.2.0.8
006b7000      4K rwx--  /lib/libtermcap.so.2.0.8
00d00000     36K r-x--  /lib/libnss_files-2.3.4.so
00d09000      4K r-x--  /lib/libnss_files-2.3.4.so
00d0a000      4K rwx--  /lib/libnss_files-2.3.4.so
08047000    580K r-x--  /bin/bash
080d8000     24K rw---  /bin/bash
080de000     20K rw---    [ anon ]
09baa000    264K rw---    [ anon ]
b7d1f000      8K rw---    [ anon ]
b7d21000     24K r--s-  /usr/lib/gconv/gconv-modules
b7d27000   2048K r----  /usr/lib/locale/locale-archive
b7f27000      8K rw---    [ anon ]
bff20000    896K rw---    [ stack ]
ffffe000      4K -----    [ anon ]
 total     5248K
```

```
$ pmap 3998
3998:   -bash
0043e000     88K r-x--  /lib/ld-2.3.4.so
00454000      4K r-x--  /lib/ld-2.3.4.so
00455000      4K rwx--  /lib/ld-2.3.4.so
00458000   1176K r-x--  /lib/tls/libc-2.3.4.so
0057e000      8K r-x--  /lib/tls/libc-2.3.4.so
00580000      8K rwx--  /lib/tls/libc-2.3.4.so
00582000      8K rwx--    [ anon ]
005a0000     36K r-x--  /lib/libnss_files-2.3.4.so
005a9000      4K r-x--  /lib/libnss_files-2.3.4.so
005aa000      4K rwx--  /lib/libnss_files-2.3.4.so
005ab000      8K r-x--  /lib/libdl-2.3.4.so
005ad000      4K r-x--  /lib/libdl-2.3.4.so
005ae000      4K rwx--  /lib/libdl-2.3.4.so
006b4000     12K r-x--  /lib/libtermcap.so.2.0.8
006b7000      4K rwx--  /lib/libtermcap.so.2.0.8
08047000    580K r-x--  /bin/bash
080d8000     24K rw---  /bin/bash
080de000     20K rw---    [ anon ]
08fae000    264K rw---    [ anon ]
b7d1b000      8K rw---    [ anon ]
b7d1d000     24K r--s-  /usr/lib/gconv/gconv-modules
b7d23000   2048K r----  /usr/lib/locale/locale-archive
b7f23000      8K rw---    [ anon ]
bfeb0000   1344K rw---    [ stack ]
ffffe000      4K -----    [ anon ]
 total     5696K
```

The yellow parts are all non-writable (shared)

How much memory does my process take?

```
$ ps -ofname,pid,vsize,rss
COMMAND     PID   VSZ  RSS
bash       3998  5692 1136
```

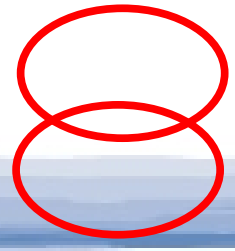
```
$ ps -ofname,pid,vsize,rss
COMMAND     PID   VSZ  RSS
bash       4027  5244 1188
```

So these two bash processes each take about 1100K of **physical** RAM

How much memory does my process take?

```
$ ps -ofname,pid,vsize,rss
COMMAND     PID   VSZ  RSS
bash       3998  5692 1136
```

```
$ ps -ofname,pid,vsize,rss
COMMAND     PID   VSZ  RSS
bash       4027  5244 1188
```



So these two bash processes each take about 1100K of **physical** RAM

BUT IT OVERLAPS TO SOME DEGREE

How much memory does my process take?

So, it's tricky.

A big VSIZE doesn't mean much.
A big RSS doesn't mean much.

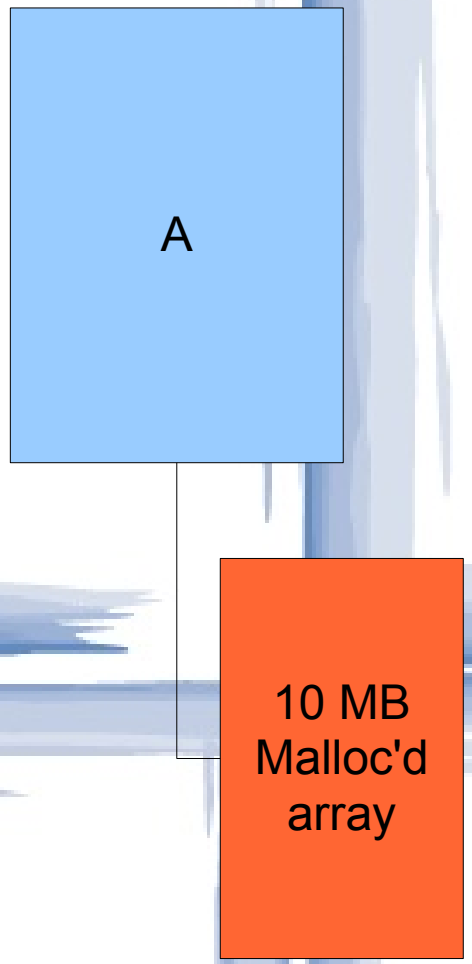
How much memory does my process take?

So, it's tricky.

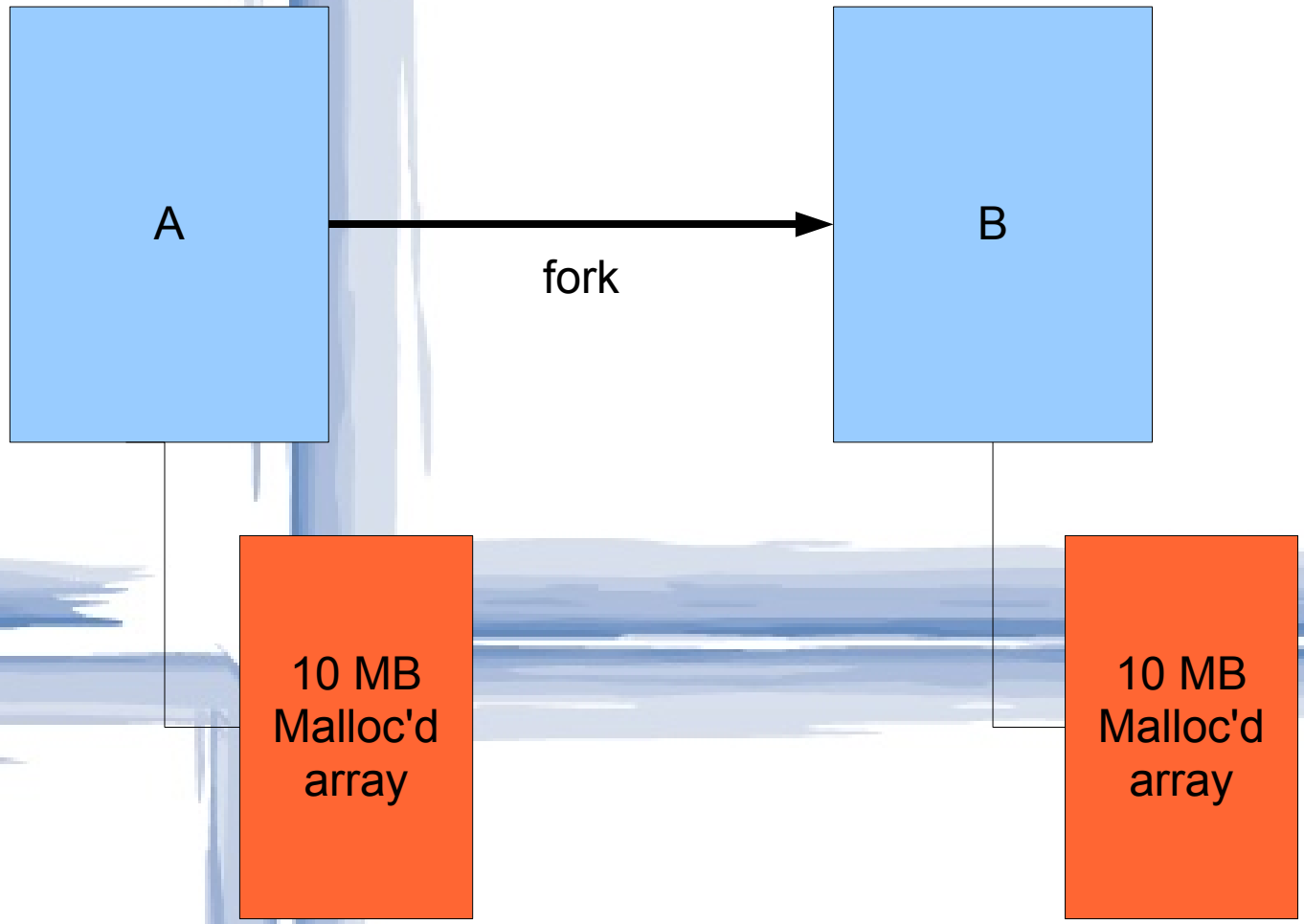
A big VSIZE doesn't mean much.
A big RSS doesn't mean much.

Overall, it's easier to gauge your system's capacity by watching overall free memory and free swap (using top), and page-in/out (using vmstat).

Copy-on-write (COW)

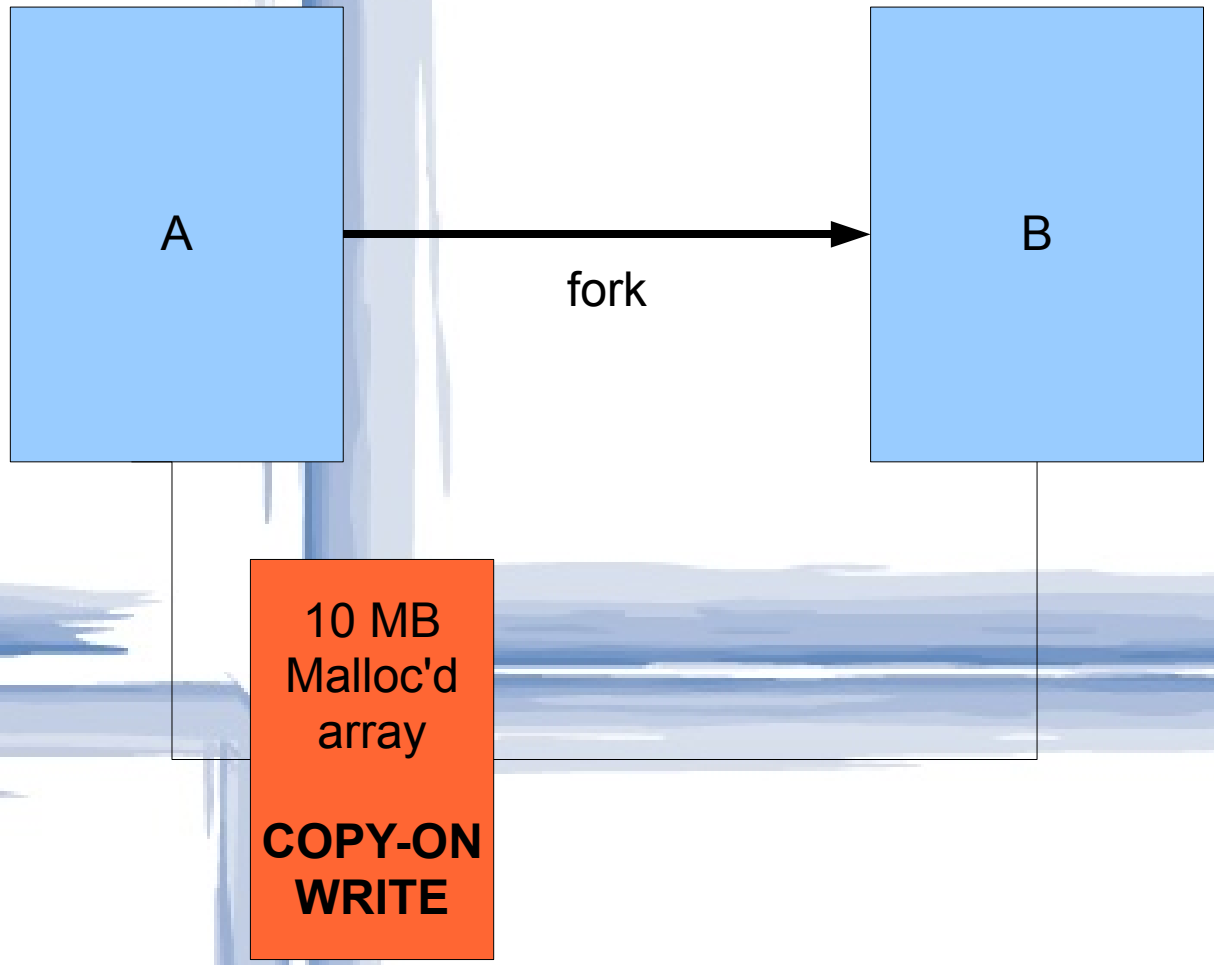


Copy-on-write (COW)



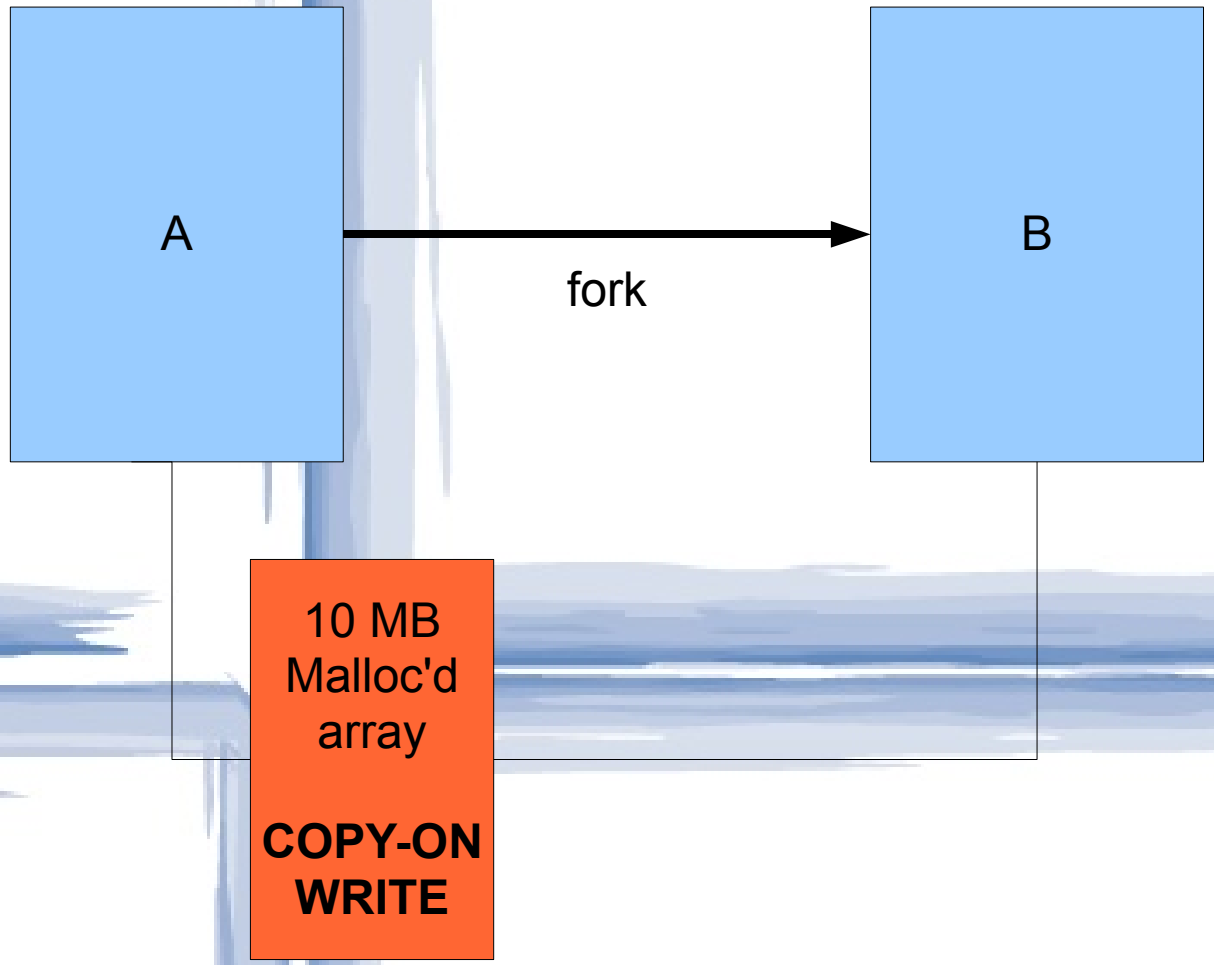
Without COW, a fork would **copy** all the parent's writable pages

Copy-on-write (COW)



With COW, a fork just marks parent's writable pages as *“copy-on-write”*

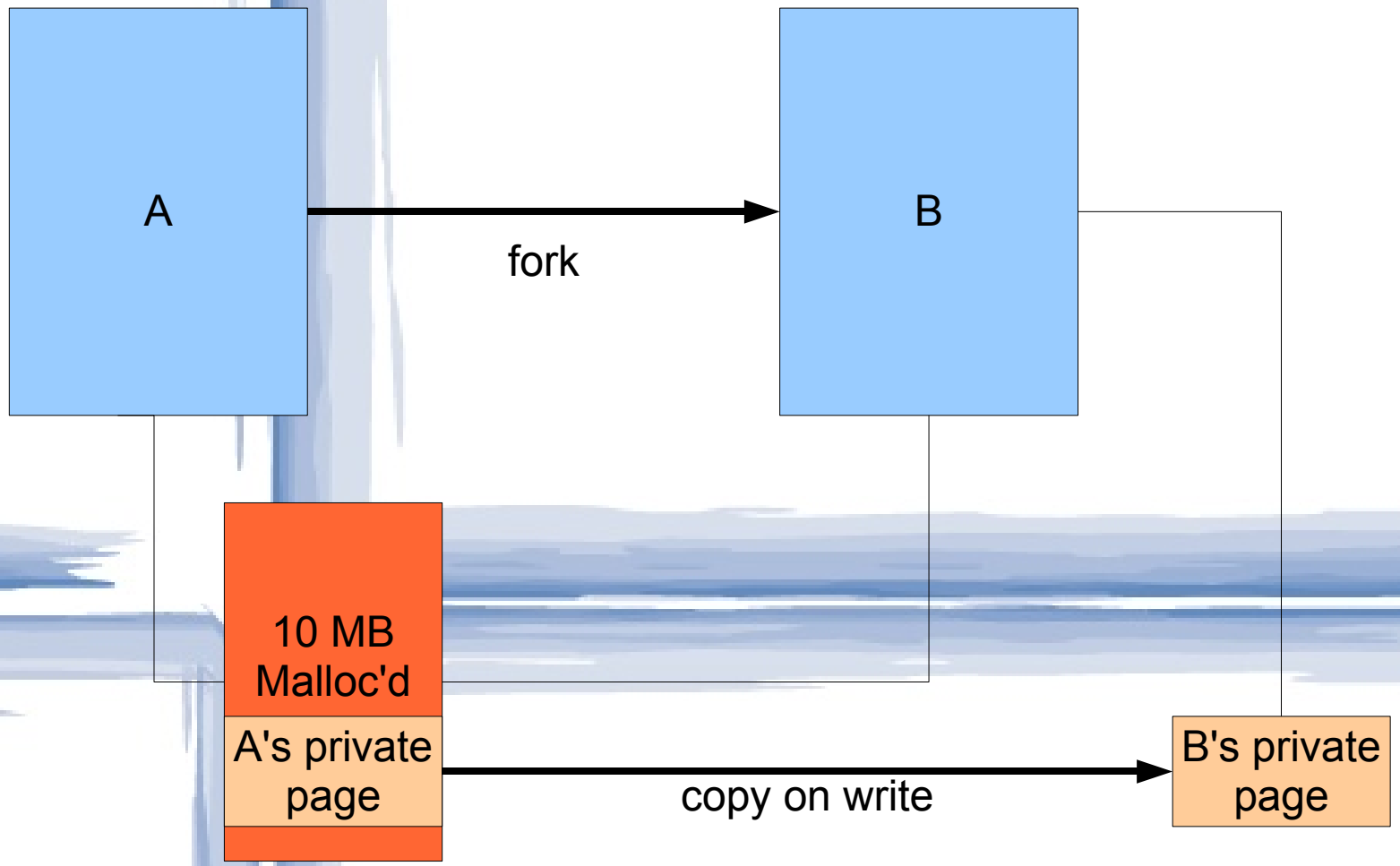
Copy-on-write (COW)



Advantages of COW:

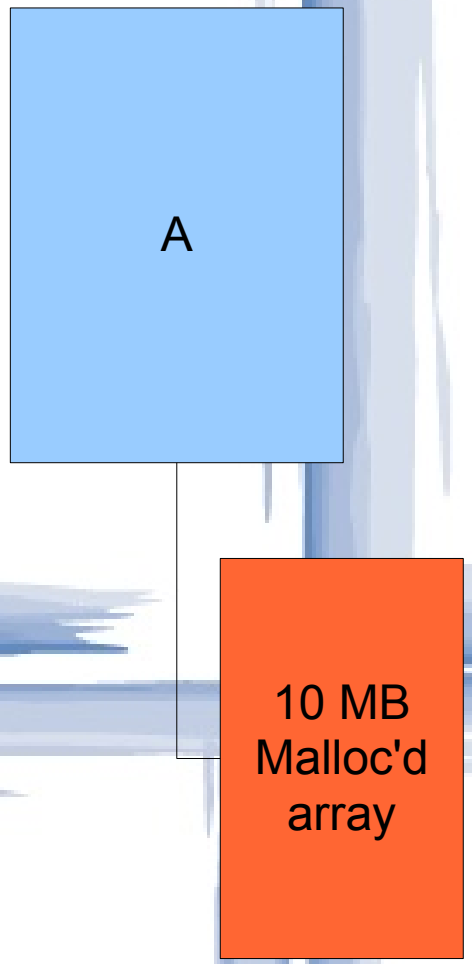
fork was **fast**, and B's array took **zero add'l** memory

Copy-on-write (COW)



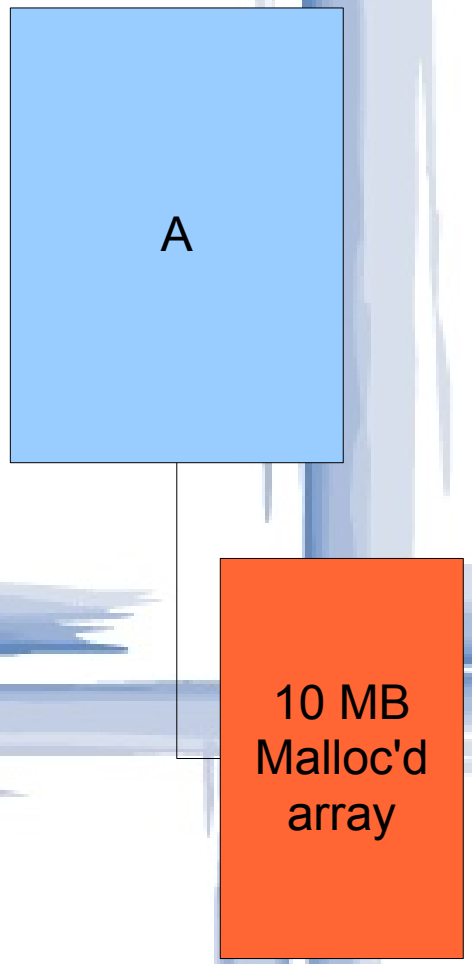
When B writes to a page, that page (alone) is copied

Example: Demand Paging & COW




```
int main() {
  int i;
  char *a;

  a = malloc(BUFSZ);
  printf("memory allocated but not touched...\n");
  sleep(10);

  for(i=0; i<BUFSZ; i++) a[i]=(char)(i%256);
  printf("memory touched...\n");
  sleep(10);

  if (fork() == 0) {
    printf("child has untouched cow pages\n");
    sleep(10);
    printf("child writing to cow pages\n");
    for(i=0; i<BUFSZ; i++) a[i]=(char)(i%10);
    sleep(10);
  } else {
    sleep(20);
  }
}
```

Example: Demand Paging & COW




```
int main() {
  int i;
  char *a;

  a = malloc(BUFSZ);
  printf("memory allocated but not touched...\n");
  sleep(10);

  for(i=0; i<BUFSZ; i++) a[i]=(char)(i%256);
  printf("memory touched...\n");
  sleep(10);

  if (fork() == 0) {
    printf("child has untouched cow pages\n");
    sleep(10);
    printf("child writing to cow pages\n");
    for(i=0; i<BUFSZ; i++) a[i]=(char)(i%10);
    sleep(10);
  } else {
    sleep(20);
  }
}
```

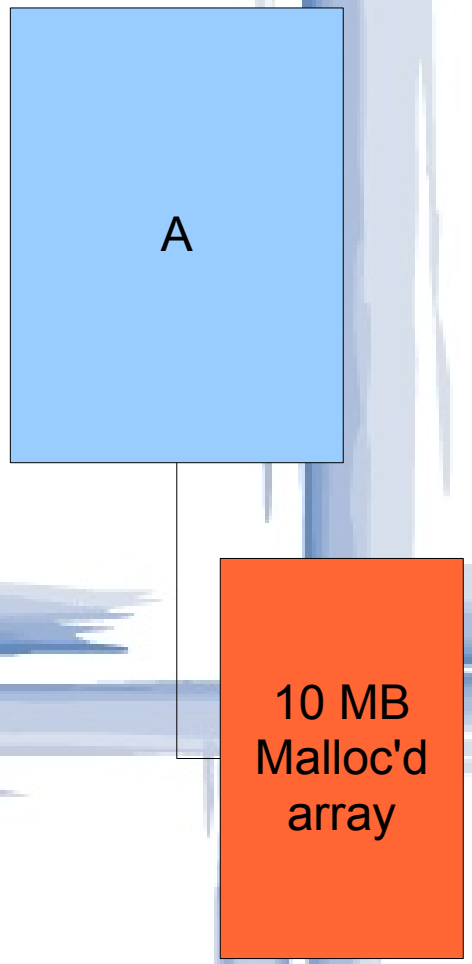
Example: Demand Paging & COW

```
int main() {
  int i;
```

```
% pmap -x 19112
19112:  ./cow
 Address  Kbytes     RSS    Anon  Locked Mode   Mapped File
08045000      12      12      12       - rwx--    [ stack ]
08050000       4       4       -       - r-x--  cow
08060000       4       4       4       - rwx--  cow
08061000   10248       8       8       - rwx--    [ heap ]
CFEC0000       4       4       4       - rwx--    [ anon ]
CFED0000     740     740       -       - r-x--  libc.so.1
CFF99000      24      24      24       - rw---  libc.so.1
CFF9F000       8       8       8       - rw---  libc.so.1
CFFB0000      24      12      12       - rwx--    [ anon ]
CFFC0000       4       4       -       - r--s-  dev:102,3
ino:1068
CFFCA000     132     132       -       - r-x--  ld.so.1
CFFFB000       4       4       4       - rwx--  ld.so.1
CFFFC000       8       8       8       - rwx--  ld.so.1
-------- ------- ------- ------- -------
total Kb   11216     964      84       -
```

```
}
```

Example: Demand Paging & COW



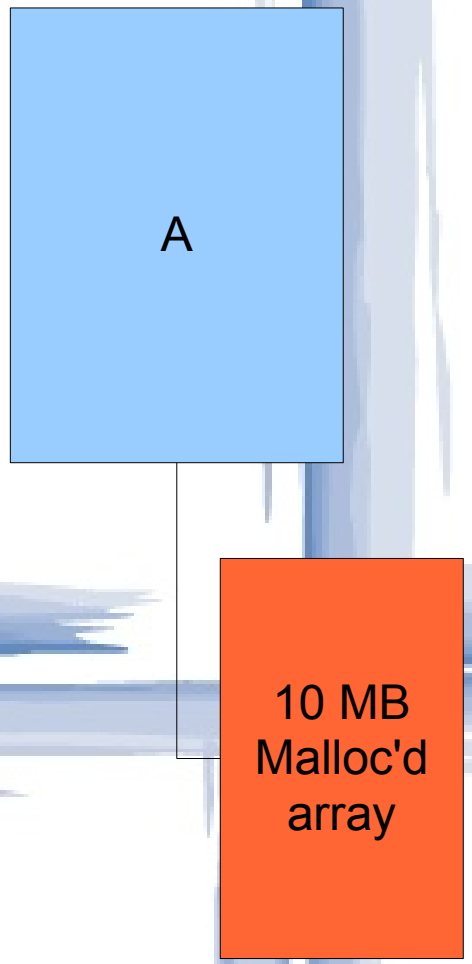

```
int main() {
  int i;
  char *a;

  a = malloc(BUFSZ);
  printf("memory allocated but not touched...\n");
  sleep(10);

  for(i=0; i<BUFSZ; i++) a[i]=(char)(i%256);
  printf("memory touched...\n");
  sleep(10);

  if (fork() == 0) {
    printf("child has untouched cow pages\n");
    sleep(10);
    printf("child writing to cow pages\n");
    for(i=0; i<BUFSZ; i++) a[i]=(char)(i%10);
    sleep(10);
  } else {
    sleep(20);
  }
}
```

Example: Demand Paging & COW

```
int main() {
  int i;
```

```
% pmap -x 19112
19112:  ./cow
 Address  Kbytes     RSS    Anon  Locked Mode   Mapped File
08045000      12      12      12       - rwx--    [ stack ]
08050000       4       4       -       - r-x--  cow
08060000       4       4       4       - rwx--  cow
08061000   10248   10244   10244       - rwx--    [ heap ]
CFEC0000       4       4       4       - rwx--    [ anon ]
CFED0000     740     740       -       - r-x--  libc.so.1
CFF99000      24      24      24       - rw---  libc.so.1
CFF9F000       8       8       8       - rw---  libc.so.1
CFFB0000      24      12      12       - rwx--    [ anon ]
CFFC0000       4       4       -       - r--s-  dev:102,3
ino:1068
CFFCA000     132     132       -       - r-x--  ld.so.1
CFFFB000       4       4       4       - rwx--  ld.so.1
CFFFC000       8       8       8       - rwx--  ld.so.1
-------- ------- ------- ------- -------
total Kb   11216   11200   10320
```

Example: Demand Paging & COW



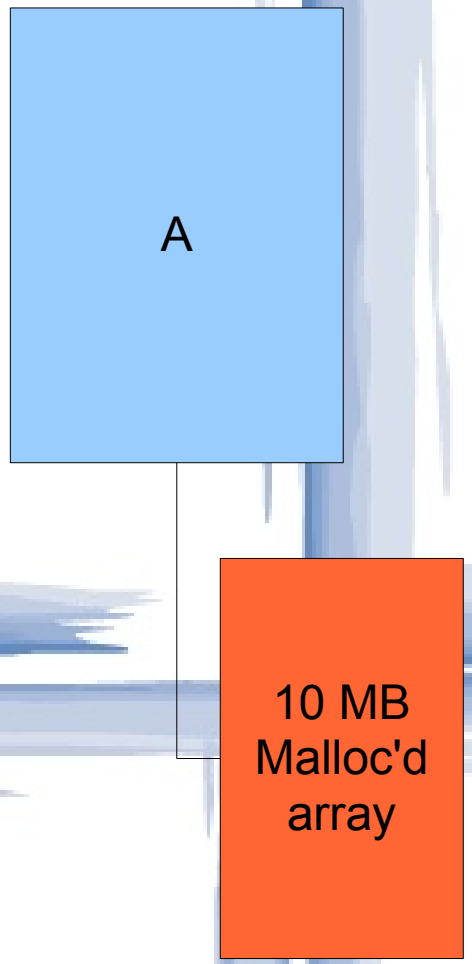

```
int main() {
  int i;
  char *a;

  a = malloc(BUFSZ);
  printf("memory allocated but not touched...\n");
  sleep(10);

  for(i=0; i<BUFSZ; i++) a[i]=(char)(i%256);
  printf("memory touched...\n");
  sleep(10);

  if (fork() == 0) {
    printf("child has untouched cow pages\n");
    sleep(10);
    printf("child writing to cow pages\n");
    for(i=0; i<BUFSZ; i++) a[i]=(char)(i%10);
    sleep(10);
  } else {
    sleep(20);
  }
}
```

Example: Demand Paging & COW

```
int main() {
  int i;
```

```
% pmap -x 19115
19115:  ./cow
 Address  Kbytes     RSS    Anon  Locked Mode   Mapped File
08045000      12      12       8       - rwx--    [ stack ]
08050000       4       4       -       - r-x--  cow
08060000       4       4       -       - rwx--  cow
08061000   10248   10244       -       - rwx--    [ heap ]
CFEC0000       4       4       -       - rwx--    [ anon ]
CFED0000     740     740       -       - r-x--  libc.so.1
CFF99000      24      24      12       - rw---  libc.so.1
CFF9F000       8       8       4       - rw---  libc.so.1
CFFB0000      24      12       4       - rwx--    [ anon ]
CFFC0000       4       4       -       - r--s-  dev:102,3
ino:1068
CFFCA000     132     132       -       - r-x--  ld.so.1
CFFFB000       4       4       4       - rwx--  ld.so.1
CFFFC000       8       8       4       - rwx--  ld.so.1
-------- ------- ------- ------- -------
total Kb   11216   11200      36       -
```

Example: Demand Paging & COW



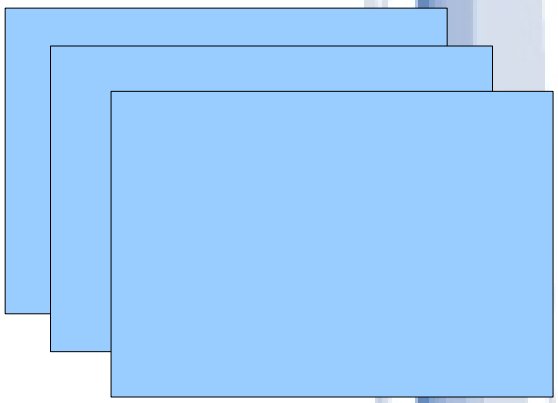

```
int main() {
  int i;
  char *a;

  a = malloc(BUFSZ);
  printf("memory allocated but not touched...\n");
  sleep(10);

  for(i=0; i<BUFSZ; i++) a[i]=(char)(i%256);
  printf("memory touched...\n");
  sleep(10);

  if (fork() == 0) {
    printf("child has untouched cow pages\n");
    sleep(10);
    printf("child writing to cow pages\n");
    for(i=0; i<BUFSZ; i++) a[i]=(char)(i%10);
    sleep(10);
  } else {
    sleep(20);
  }
}
```

Example: Demand Paging & COW

```
int main() {
  int i;
```

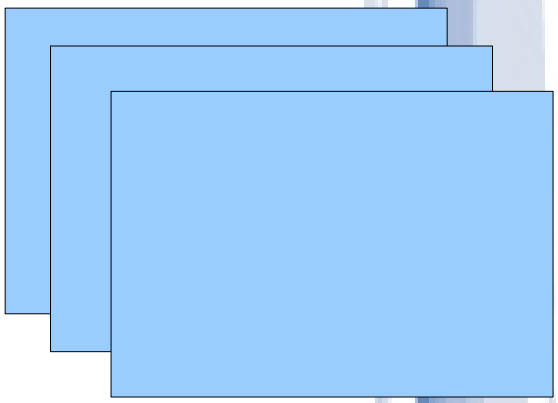
```
% pmap -x 19115
19115:  ./cow
 Address  Kbytes     RSS    Anon  Locked Mode   Mapped File
08045000      12      12       8       - rwx--    [ stack ]
08050000       4       4       -       - r-x--  cow
08060000       4       4       4       - rwx--  cow
08061000   10248   10244   10240       - rwx--    [ heap ]
CFEC0000       4       4       -       - rwx--    [ anon ]
CFED0000     740     740       -       - r-x--  libc.so.1
CFF99000      24      24      12       - rw---  libc.so.1
CFF9F000       8       8       4       - rw---  libc.so.1
CFFB0000      24      12       4       - rwx--    [ anon ]
CFFC0000       4       4       -       - r--s-  dev:102,3
ino:1068
CFFCA000     132     132       -       - r-x--  ld.so.1
CFFFB000       4       4       4       - rwx--  ld.so.1
CFFFC000       8       8       4       - rwx--  ld.so.1
-------- ------- ------- ------- -------
total Kb   11216   11200   10280       -
```

Overcommit



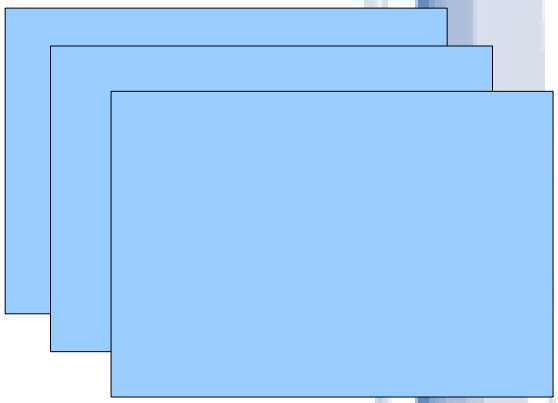
Sum of all processes VSIZE **>** total RAM + swap

Overcommit



- Sum of all processes VSIZE **>** total RAM + swap
-
  - **Normally this is OK!**
- Processes don't really use all the memory they allocate
- Many processes share their parents memory due to COW

Overcommit

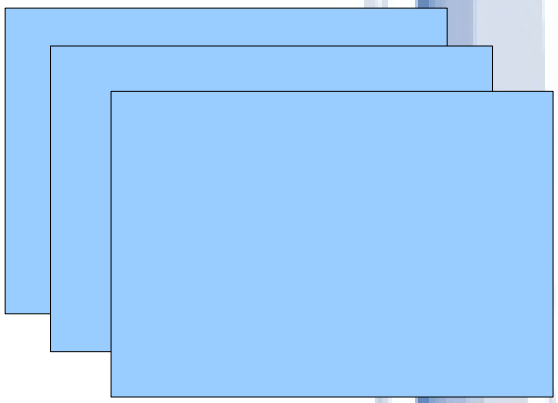


Sum of all processes VSIZE **>** total RAM + swap

You can shut overcommit off:

```
echo 2 > /proc/sys/vm/overcommit_memory
echo 50 > /proc/sys/vm/overcommit_ratio
```

Overcommit



Sum of all processes VSIZE **>** total RAM + swap

When overcommit fails

mallocs succeeded.... forks succeeded....

But when you actually tried to use all your pages, the kernel could not deliver on its promise!

The OOM killer

Running out of memory

```
Mar 24 12:40:02 linux02 kernel: oom-killer
Mar 24 12:40:06 linux02 kernel: Free pages:        1032kB (112kB HighMem)
Mar 24 12:40:09 linux02 kernel: Active:98710 inactive:103620 dirty:0
writeback:0 unstable:0 free:258 slab:3217 mapped:202100 pagetables:1044
Mar 24 12:40:10 linux02 kernel: DMA free:16kB min:16kB low:32kB high:48kB
active:7384kB inactive:5052kB present:16384kB pages_scanned:13794
all_unreclaimable? yes
Mar 24 12:40:12 linux02 kernel: Normal free:904kB min:936kB low:1872kB
high:2808kB active:289904kB inactive:380988kB present:901120kB
pages_scanned:962808 all_unreclaimable? yes
Mar 24 12:40:13 linux02 kernel: HighMem free:112kB min:128kB low:256kB
high:384kB active:97552kB inactive:28440kB present:129472kB
pages_scanned:142390 all_unreclaimable? yes
Mar 24 12:40:15 linux02 kernel: 261744 pages of RAM
Mar 24 12:40:15 linux02 kernel: 32368 pages of HIGHMEM
Mar 24 12:40:15 linux02 kernel: 52404 reserved pages
Mar 24 12:40:16 linux02 kernel: 126546 pages shared
Mar 24 12:40:16 linux02 kernel: 147061 pages swap cached
Mar 24 12:40:17 linux02 kernel: Out of Memory: Killed process 19248
(iptrap).
```

/var/log/messages

Running out of memory

```
Tasks: 106 total,  11 running,  95 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.1% us, 67.5% sy,  0.0% ni,  0.0% id, 32.2% wa,  0.2% hi,  0.0% si
Mem:   1033844k total,  1019188k used,    14656k free,      116k buffers
Swap:   524280k total,   524276k used,        4k free,   221264k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 5836 root      17  -1  356m 139m  11m R  6.8 13.9   0:01.77 iptrap
31665 tomcat    24   0  288m  20m 1912 S  5.1  2.1   0:41.28 java
 2308 fidelis   RT -10  263m 199m 192m S  0.0 19.7   0:00.48 sensor
 5654 fidelis   14  -1  188m 178m  12m S  0.0 17.7   0:03.56 mailer
 5034 fidelis   15  -1  179m 179m  12m S  0.0 17.8   0:01.29 scipd
 5026 fidelis   15  -1  178m 178m  12m S  0.0 17.7   0:01.37 icapd
 5014 fidelis   16  -1  168m 4420 1144 S  0.0  0.4   0:00.98 wratd
 2301 fidelis   15   0 54172  928  884 S  4.1  0.1   0:01.07 sysmon
 2313 fidelis   19   0 50972  49m  48m S  0.0  4.9   0:00.01 tcpkd
 2518 mysql     16   0 37684 3552  684 S 10.0  0.3   0:09.11 mysqld
 2594 fidelis   19   4 20528 1164  684 R  1.0  0.1   0:00.70 dbwriterd
```

top

This system had ~14mb free RAM and 4k free swap when OOM happened

Running out of memory

```
Tasks: 106 total,  11 running,  95 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.1% us, 67.5% sy,  0.0% ni,  0.0% id, 32.2% wa,  0.2% hi,  0.0% si
Mem:   1033844k total,  1019188k used,    14656k free,      116k buffers
Swap:   524280k total,   524276k used,        4k free,   221264k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 5836 root      17  -1  356m 139m  11m R  6.8 13.9   0:01.77 iptrap
31665 tomcat    24   0  288m  20m 1912 S  5.1  2.1   0:41.28 java
 2308 fidelis   RT -10  263m 199m 192m S  0.0 19.7   0:00.48 sensor
 5654 fidelis   14  -1  188m 178m  12m S  0.0 17.7   0:03.56 mailer
 5034 fidelis   15  -1  179m 179m  12m S  0.0 17.8   0:01.29 scipd
 5026 fidelis   15  -1  178m 178m  12m S  0.0 17.7   0:01.37 icapd
 5014 fidelis   16  -1  168m 4420 1144 S  0.0  0.4   0:00.98 wratd
 2301 fidelis   15   0 54172  928  884 S  4.1  0.1   0:01.07 sysmon
 2313 fidelis   19   0 50972  49m  48m S  0.0  4.9   0:00.01 tcpkd
 2518 mysql     16   0 37684 3552  684 S 10.0  0.3   0:09.11 mysqld
 2594 fidelis   19   4 20528 1164  684 R  1.0  0.1   0:00.70 dbwriterd
```

My system had an OOM with ~2GB free swap.. why?

Running out of memory

```
Tasks: 106 total,  11 running,  95 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.1% us, 67.5% sy,  0.0% ni,  0.0% id, 32.2% wa,  0.2% hi,  0.0% si
Mem:   1033844k total,  1019188k used,    14656k free,      116k buffers
Swap:   524280k total,   524276k used,        4k free,   221264k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 5836 root      17  -1  356m 139m  11m R  6.8 13.9   0:01.77 iptrap
31665 tomcat    24   0  288m  20m 1912 S  5.1  2.1   0:41.28 java
 2308 fidelis   RT -10  263m 199m 192m S  0.0 19.7   0:00.48 sensor
 5654 fidelis   14  -1  188m 178m  12m S  0.0 17.7   0:03.56 mailer
 5034 fidelis   15  -1  179m 179m  12m S  0.0 17.8   0:01.29 scipd
 5026 fidelis   15  -1  178m 178m  12m S  0.0 17.7   0:01.37 icapd
 5014 fidelis   16  -1  168m 4420 1144 S  0.0  0.4   0:00.98 wratd
 2301 fidelis   15   0 54172  928  884 S  4.1  0.1   0:01.07 sysmon
 2313 fidelis   19   0 50972  49m  48m S  0.0  4.9   0:00.01 tcpkd
 2518 mysql     16   0 37684 3552  684 S 10.0  0.3   0:09.11 mysqld
 2594 fidelis   19   4 20528 1164  684 R  1.0  0.1   0:00.70 dbwriterd
```

My system had an OOM with ~2GB free swap.. why?

Many fids procs use **mlockall()** -- can't be swapped!

More resources

- http://en.wikipedia.org/wiki/Virtual_memory
- **/proc/sys/vm** filesystem on Linux
- Linux Kernel Development, by Robert Love
-