Admin

- Correction: No class week 4 (not week 3 as advertised earlier)

- Note on identifying information in Assignment 1

- Cybersecurity seminars: Mondays 4-6pm IW B23

Executing Programs

Background: Operating systems

- What are the roles of the operating system?
- Present an abstraction of the hardware
- Allow sharing of the hardware
  - **Implies protection**

Computer Programs

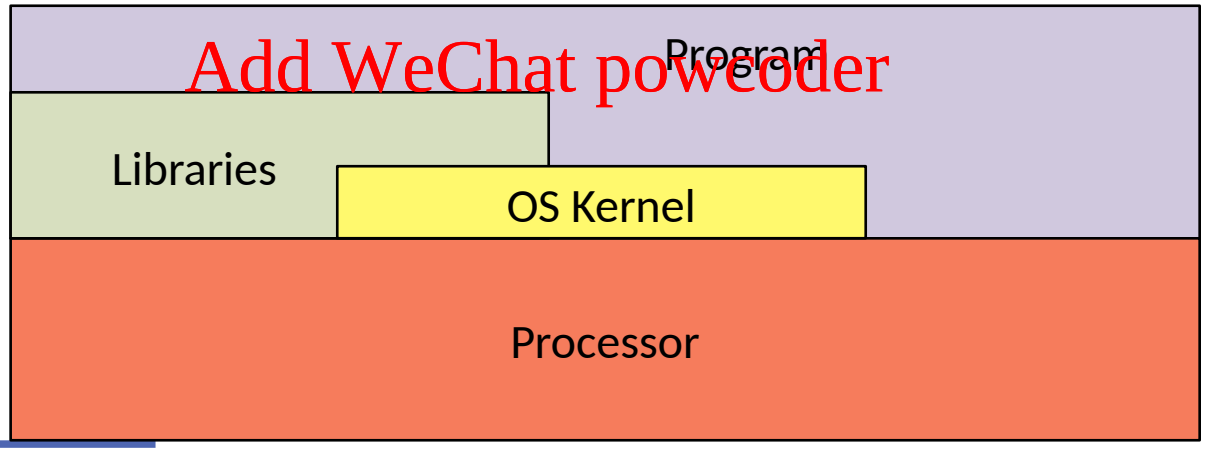
```c
#include <stdio.h>

int main(int argc, char **argv)
{
  printf("Hello World!\n");
  exit(0);
}
```


- An abstraction of a computation task
- How do we execute a program?

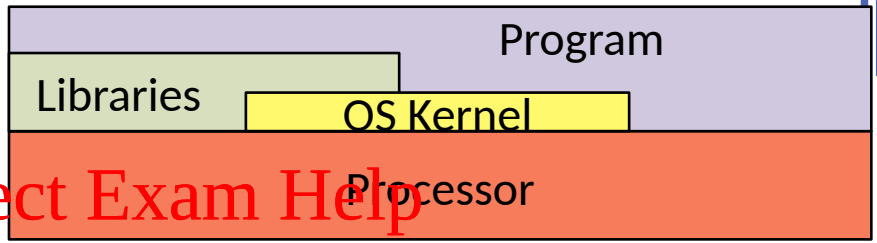
Program execution

1. Build
   - Compile – create *relocatable object* files
   - Link – create *executable* files
2. Run
   - Launch a *process*



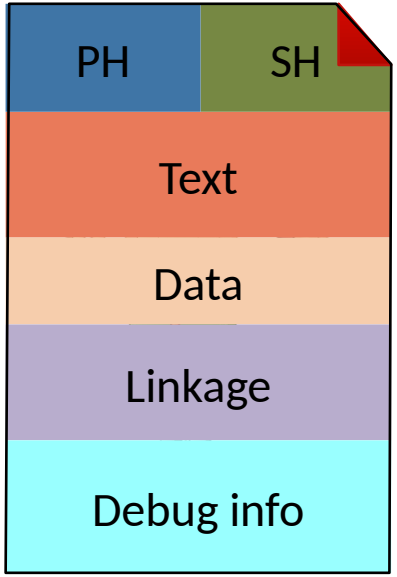
An Executable File

- A program in a form that the operating system can execute
- Specifies:
  - Machine code to be executed
  - Initial data
  - Initial memory layout
  - Libraries to link with (and required functions)
  - Debug information
  - More…



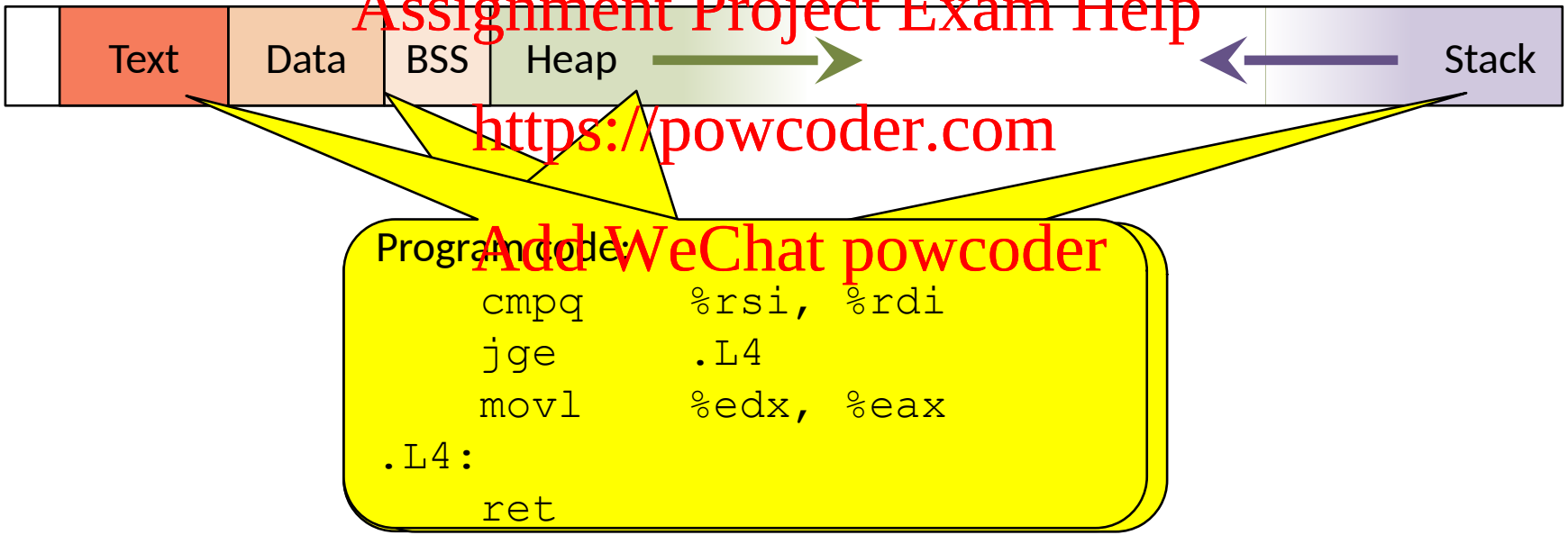
ELF
(Executable and Linkable Format)

- File format for binary programs
  - Relocatable objects
  - Executable files
  - Shared libraries
- Program header – addresses of segments to be loaded when a program is launched
- Section header – a table of ***sections***, each representing one kind of data



Memory layout

- A process executes within a *virtual address space*



Memory layout

- A process executes within a *virtual address space*


```
char str[] = "A string.";

char *toupperstr(const char *s) {
  char *res = strdup(s);
  for (char *p = res; *p; p++)
    *p = toupper(*p);
  return res;
}
```

Memory layout

- A process executes within a *virtual address space*


```
// Was: char str[] = "A string.";
char *str = "A string.";
```



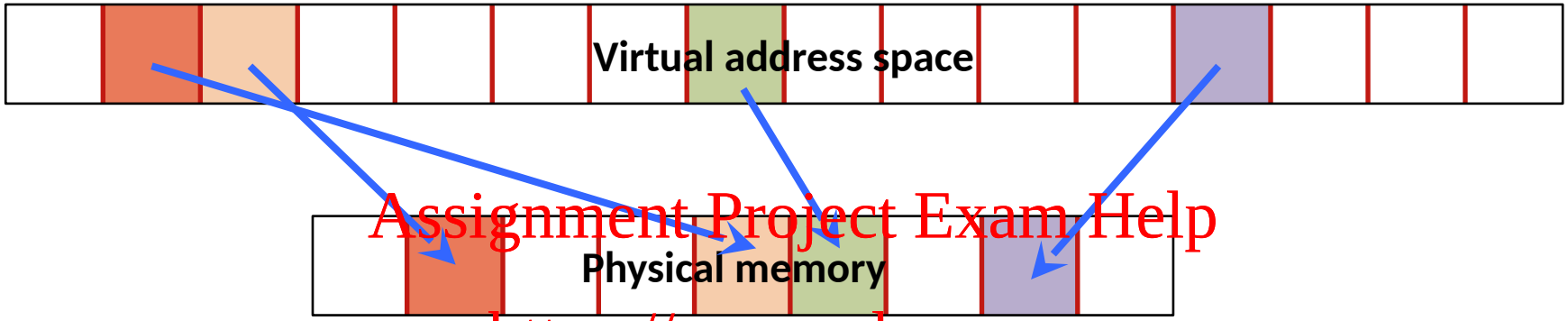
```
char *toupperstr(const char *s) {
  char *res = strdup(s);
  for (char *p = res; *p; p++)
    *p = toupper(*p);
  return res;
}
```

Sharing text

- Processes that execute the same program can share the text

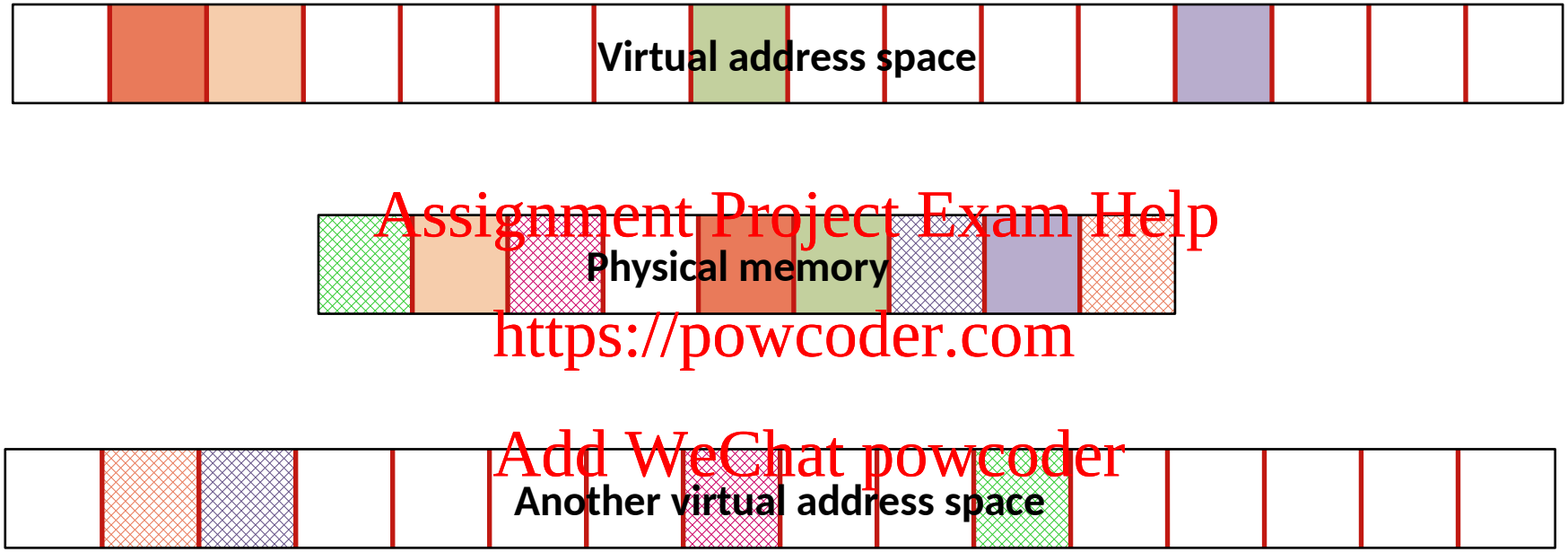
- Can also share the code of shared libraries

Implementation – Virtual memory



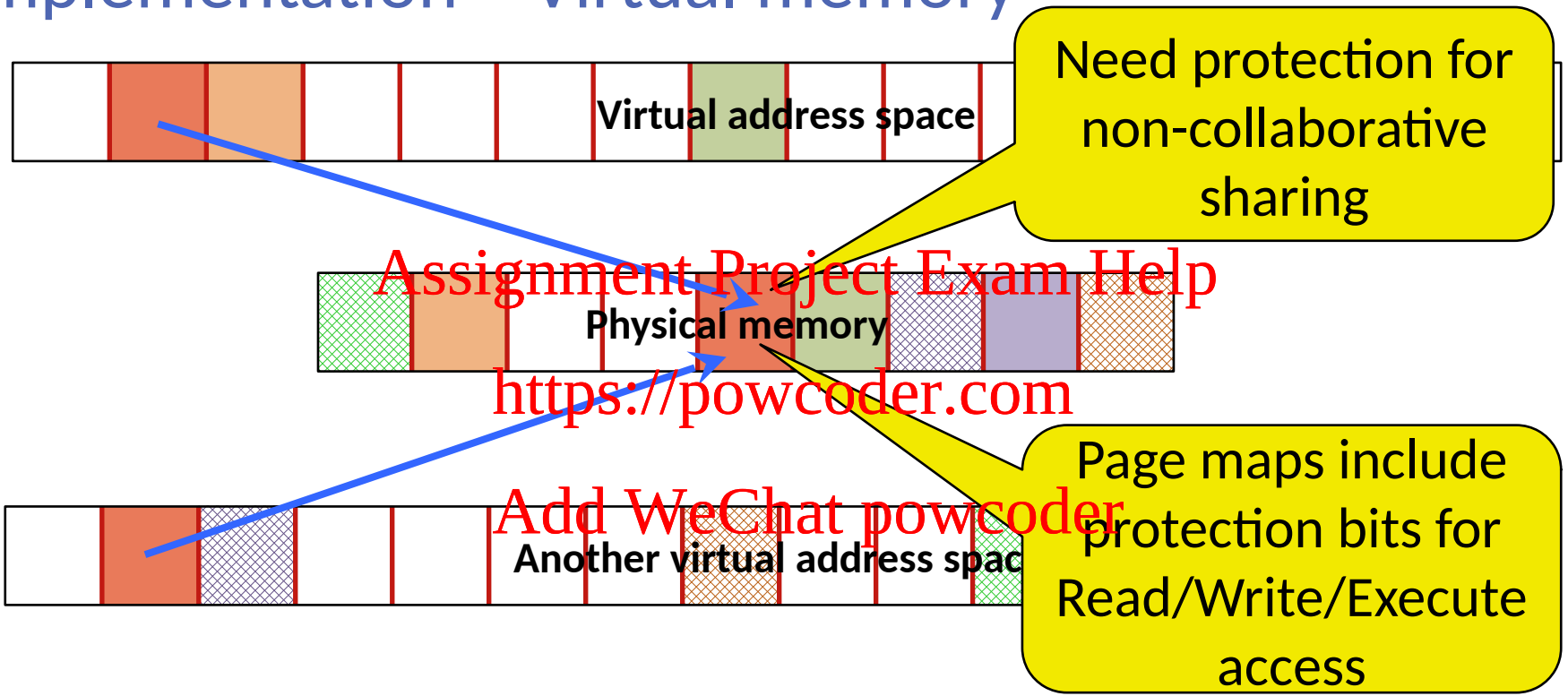
- The virtual address space is divided into fixed size ***pages***
- The physical memory is divided into page-sized ***frames***
- The OS maintains a mapping of pages to frames
- The processor uses the map to convert virtual to physical addresses

Implementation – Virtual memory



- Multiple processes can use the physical memory at the same time

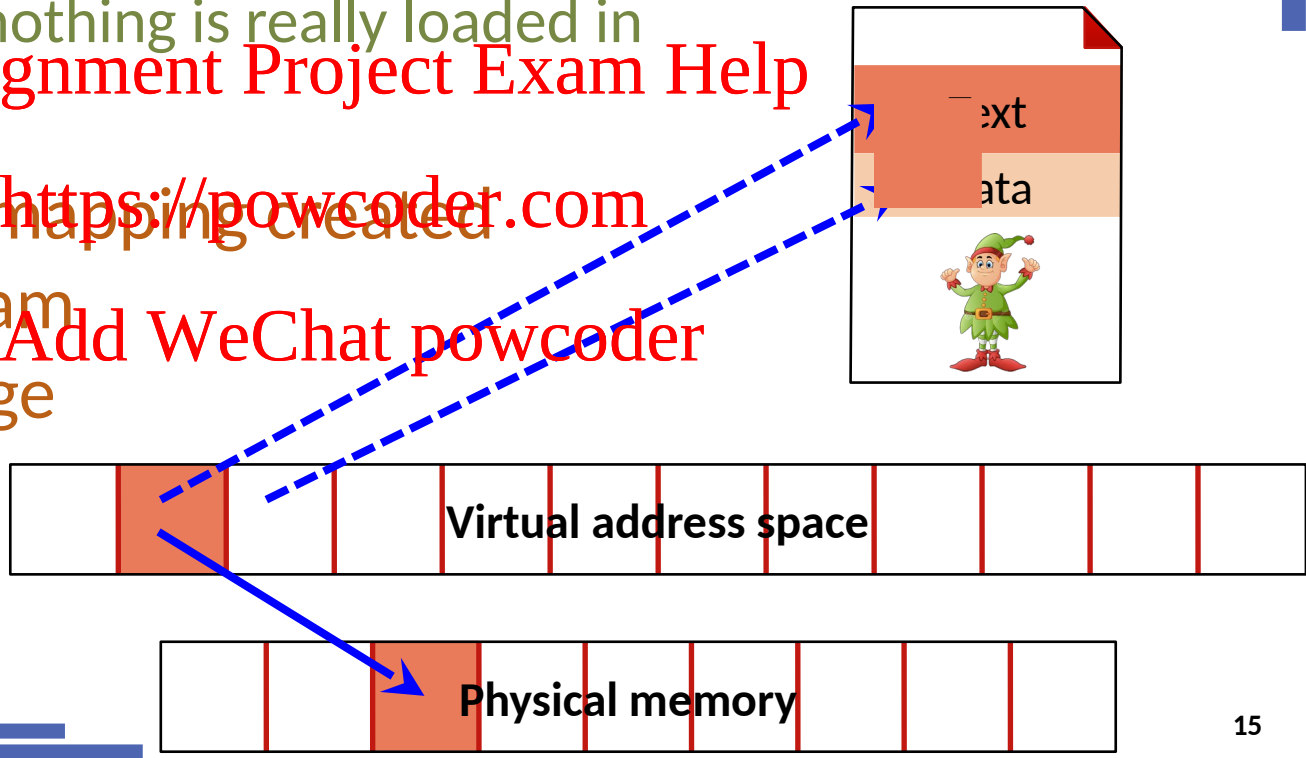
Implementation – Virtual memory



- Sharing is implemented by mapping the same physical page to multiple address spaces

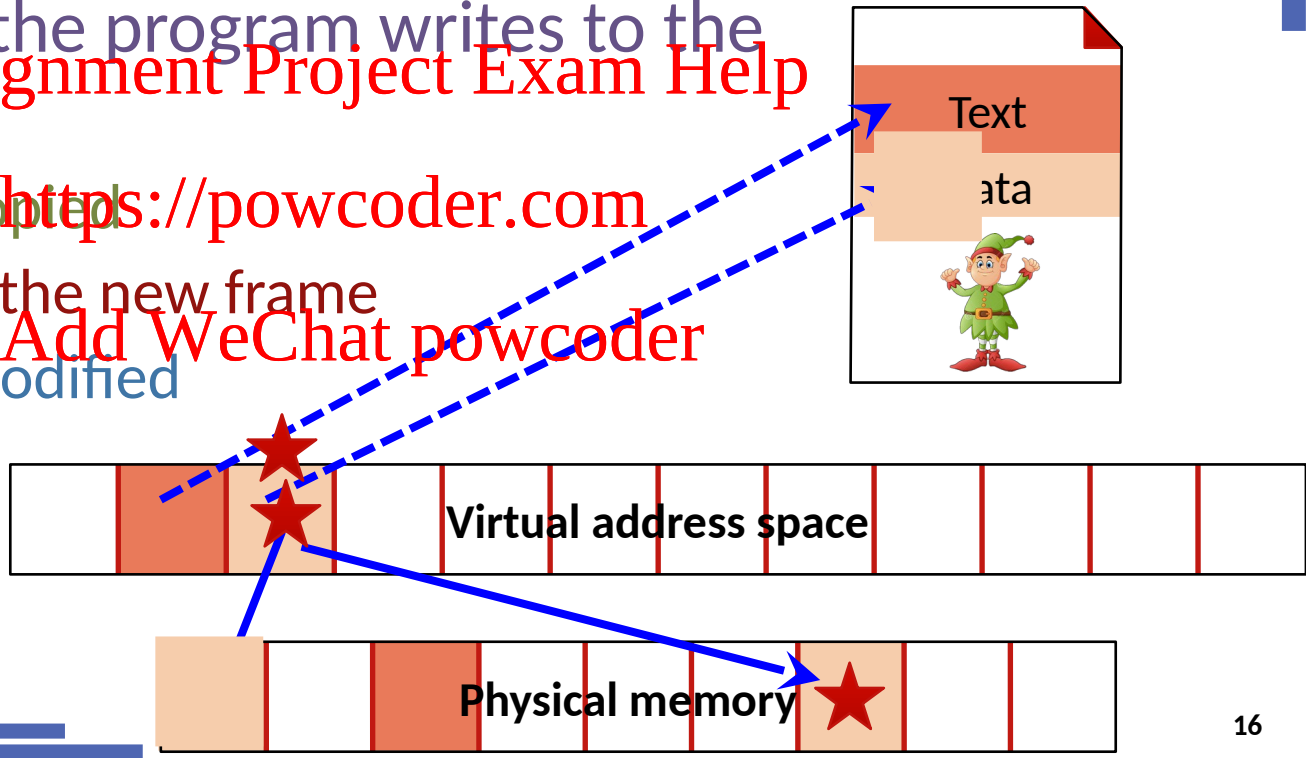
Further down the rabbit hole

- The mmap function maps a file to the virtual address space of a process.
  - Lazy mapping – nothing is really loaded in memory
- Virtual address mapping created when the program accesses the page

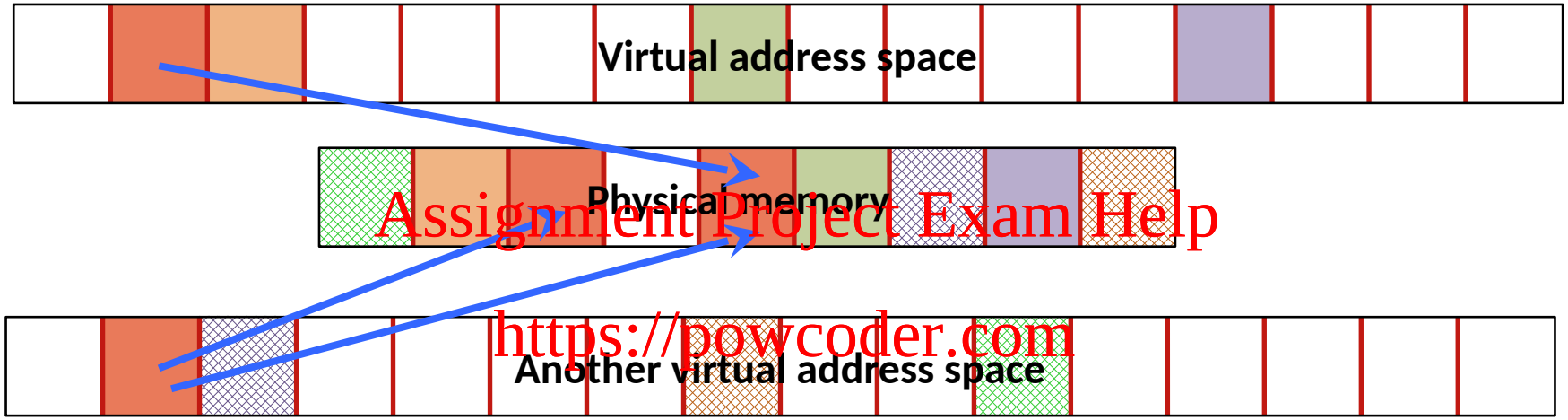


Copy-on-Write (COW)

- Data lazily loaded – marked Read Only
- The first time the program writes to the data
  - The frame is copied
  - Page points to the new frame
  - The frame is modified



Page de-duplication



- A memory-footprint-reduction technique that
  - Scans physical memory
  - Coalesces frames with identical contents
    - Uses COW to avoid interference

Class exercise

- You want to check if "Discrete Child Tracker" is installed on your phone.
- Discrete Child Tracker doesn't show up in your apps list
  - That's not surprising – that's why it's called "discrete"
- Your phone uses page de-duplication
- How can you exploit this to check if the app is installed?