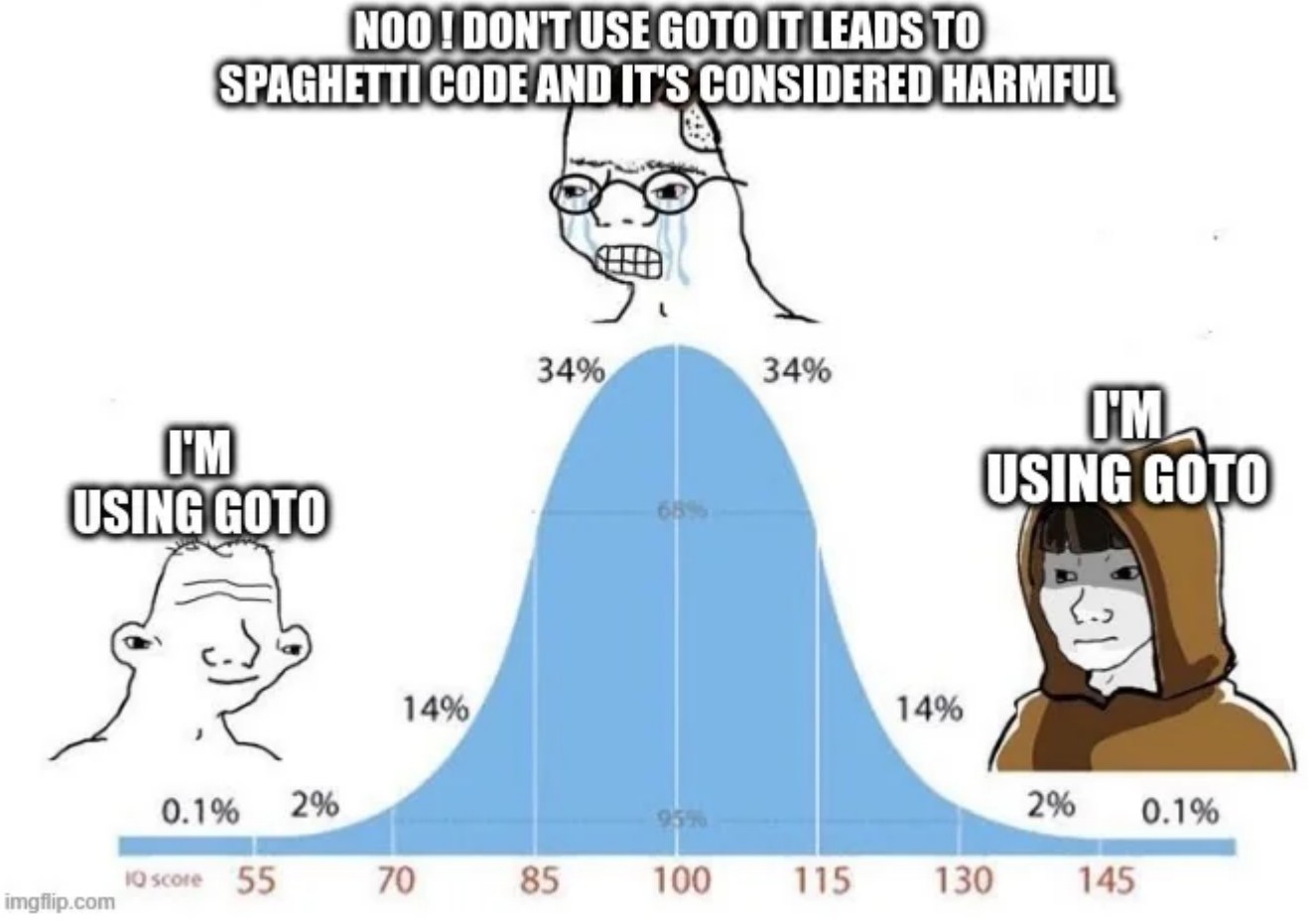

NOO ! DON'T USE GOTO IT LEADS TO
SPAGHETTI CODE AND IT'S CONSIDERED HARMFUL
I'M
USING GOTO
I'M
USING GOTO
34%
34%
14%
14%
0.1%
2%
2%
0.1%
IQ score
55
70
85
100
115
130
145
imgflip.com

TODO

- Assignment 3 assigned due October 28

Test 2 graded

- Max grade: 97.5
- A’s: 7
- B’s: 5
- C’s: 3
- Other: 2

Non-void function with one argument

```
prod10:
        movq    %rdi, %rax
        movl    $1, %ecx
        movl    $0, %esi
        jmp     .L2
.L3:    movl    (%rax), %edx
        imull   %edx, %ecx
        addq    $4, %rax
        addl    $1, %esi
.L2:    movl    %esi, %edi
        cmpl    $9, %edi
        jle     .L3             # jump if less or equal to
        movl    %ecx, %eax
        ret
```

Overview for today

- Lab 3
- Using GDB without source code

Lab 3: BitBombs

- Personalized lab
- Everyone gets a unique assignment
- Find assignment link on `assignments/lab3` on `krebs`
- Group work: pick any of your assignments

Lab 3: Studying proprietary software

- We will study it by *reverse engineering* it – attempting to reconstruct what a piece of software is doing
- Why is reverse engineering proprietary software important?

Proprietary software

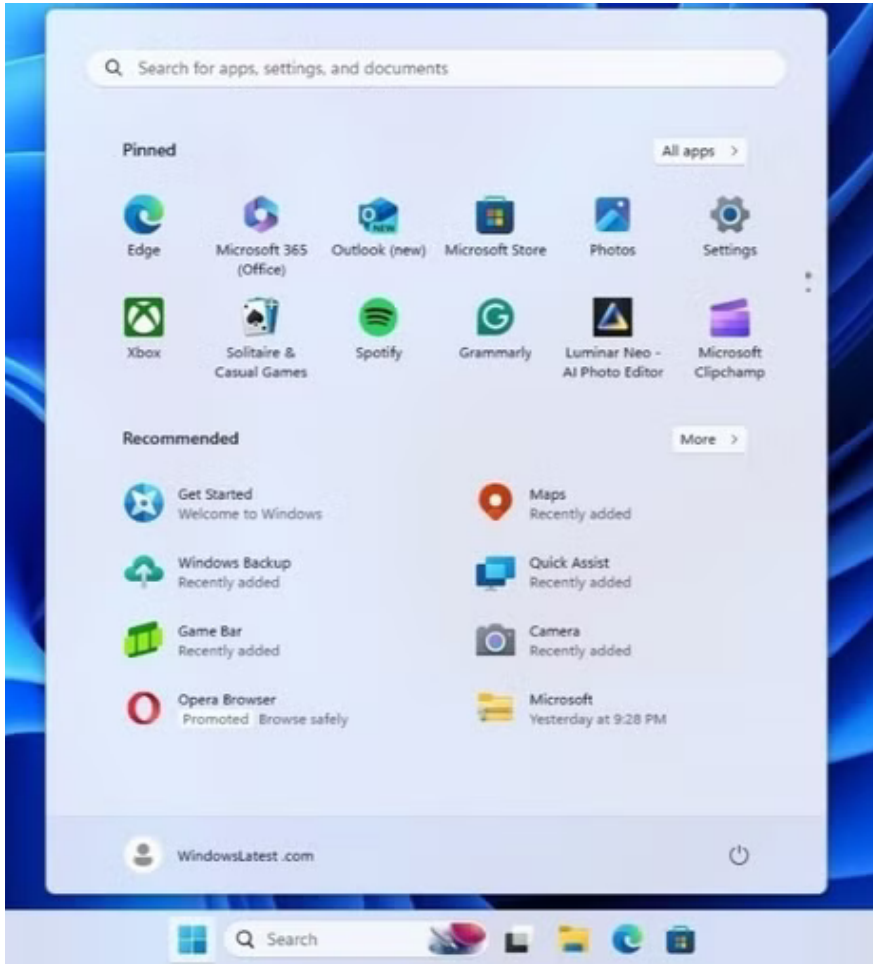
- Closed source – the software is distributed to you in compiled form without the source code
- Distributed by large megacorporations to make them as much money as possible
- Contain ads, tracking, surveillance, nags, subscriptions, and other pathological behavior
  - Not because it is what's best for you
  - But because it is best for them

Open source software

- Open source software – software designed by you and me for you and me
- We make the source code available to each other so that we can mutually benefit from the features and bug fixes that we contribute
- We don’t add ads, tracking, surveillance, nags, subscriptions, artificial restrictions, or other pathological behavior because we design this software for ourselves

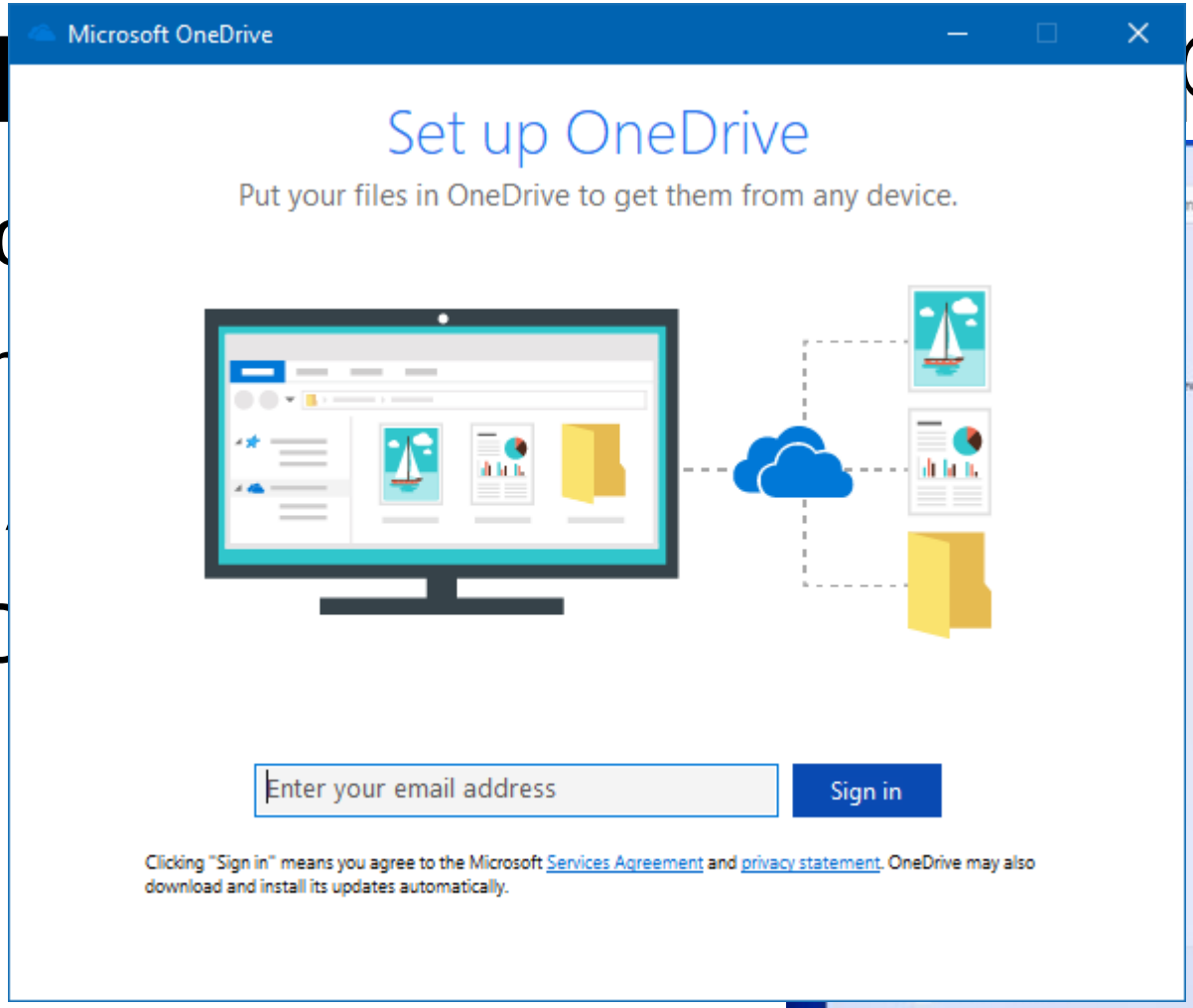
Proprietary software examples

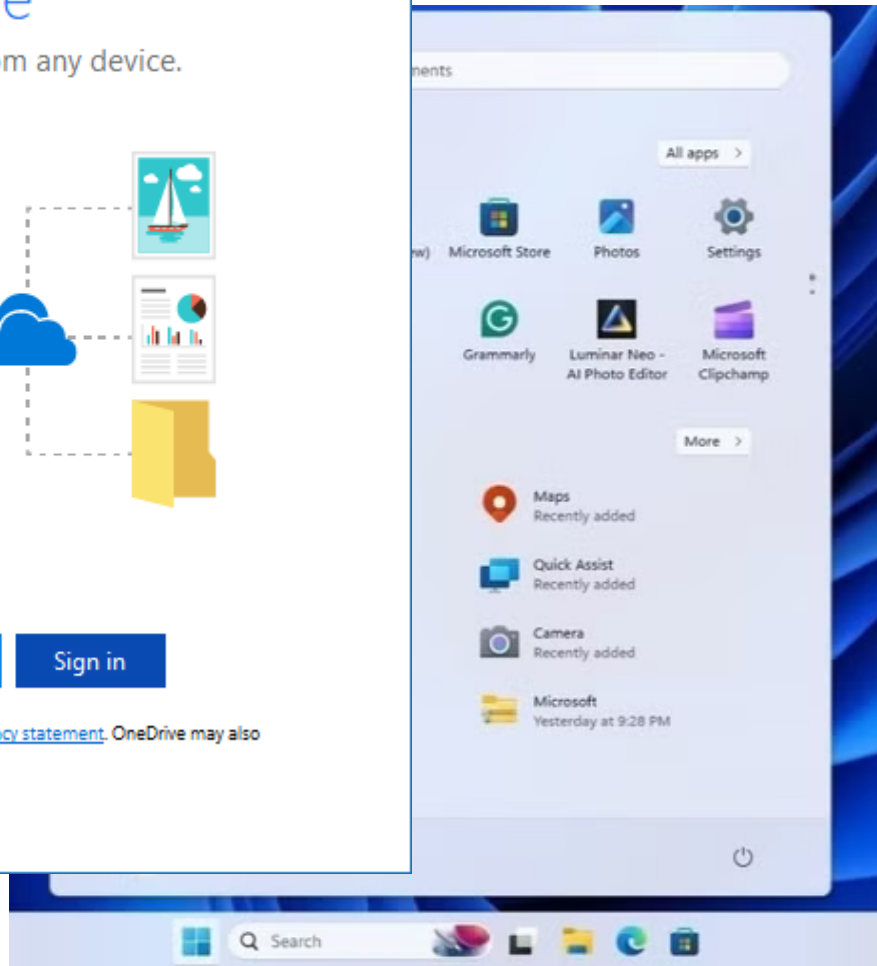
- Microsoft Windows
- Apple macOS
- Adobe Acrobat Reader
- PaperCut



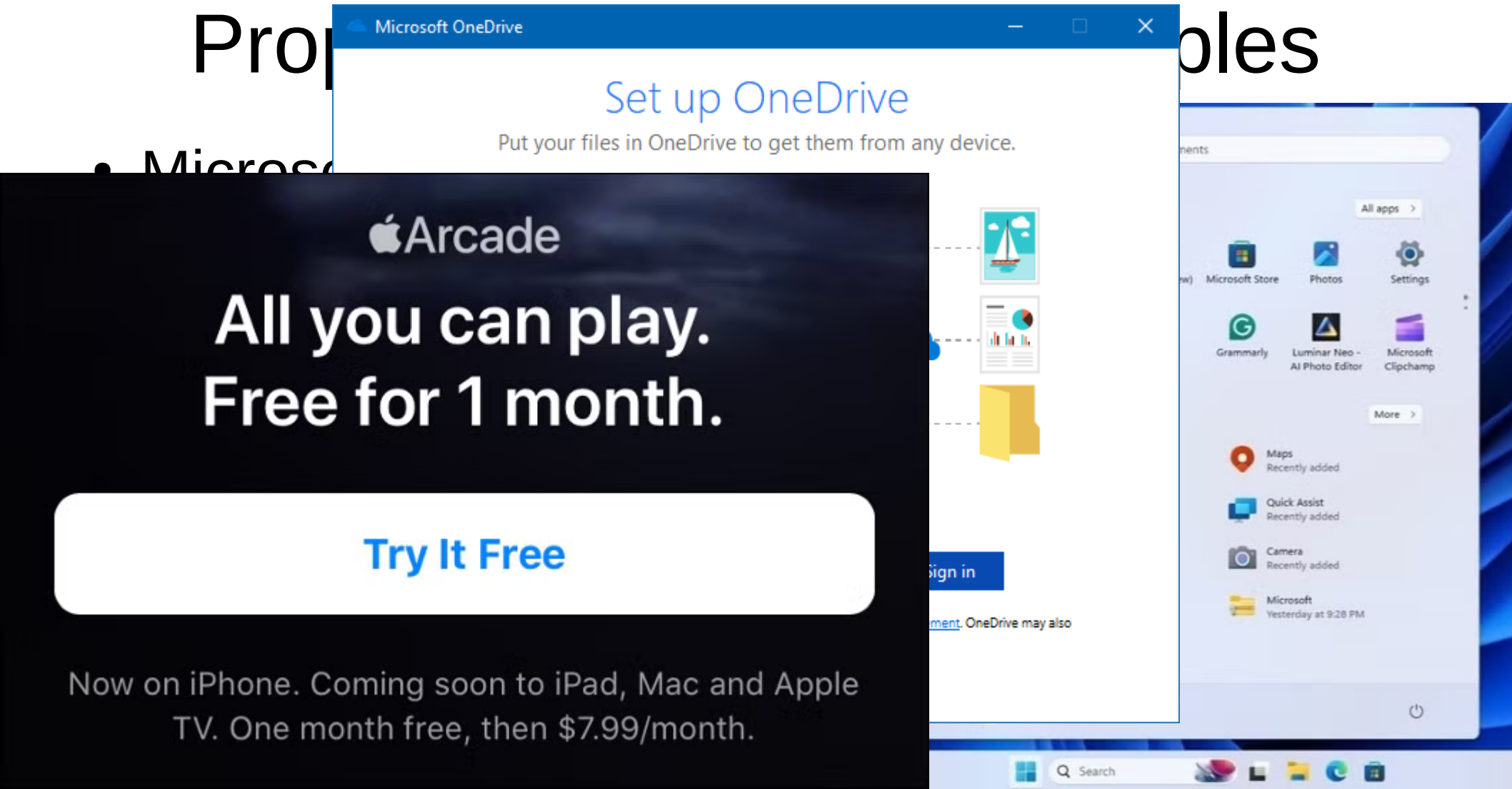
Prop ples

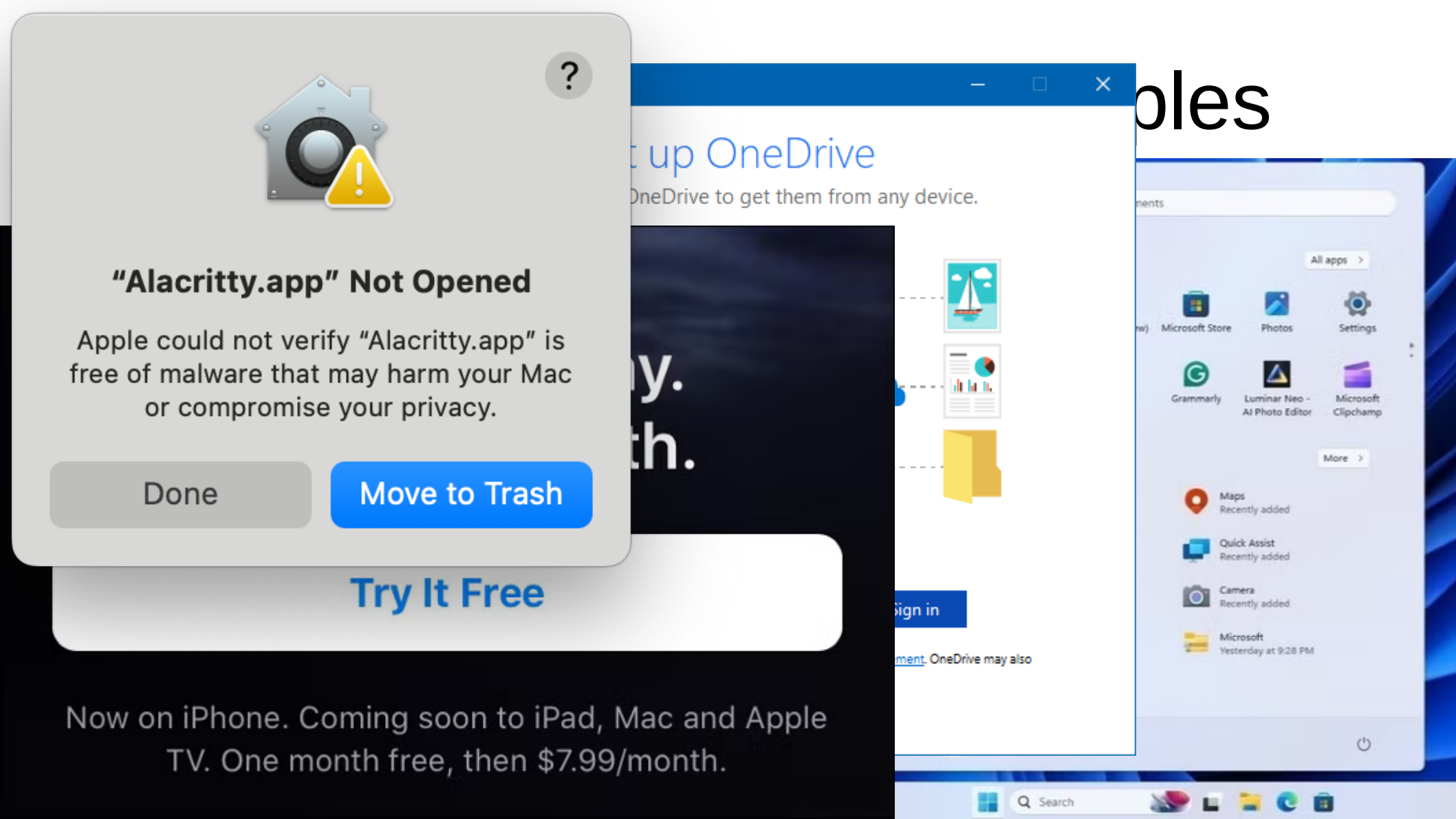
- Microso
- Apple r
- Adobe
- PaperC

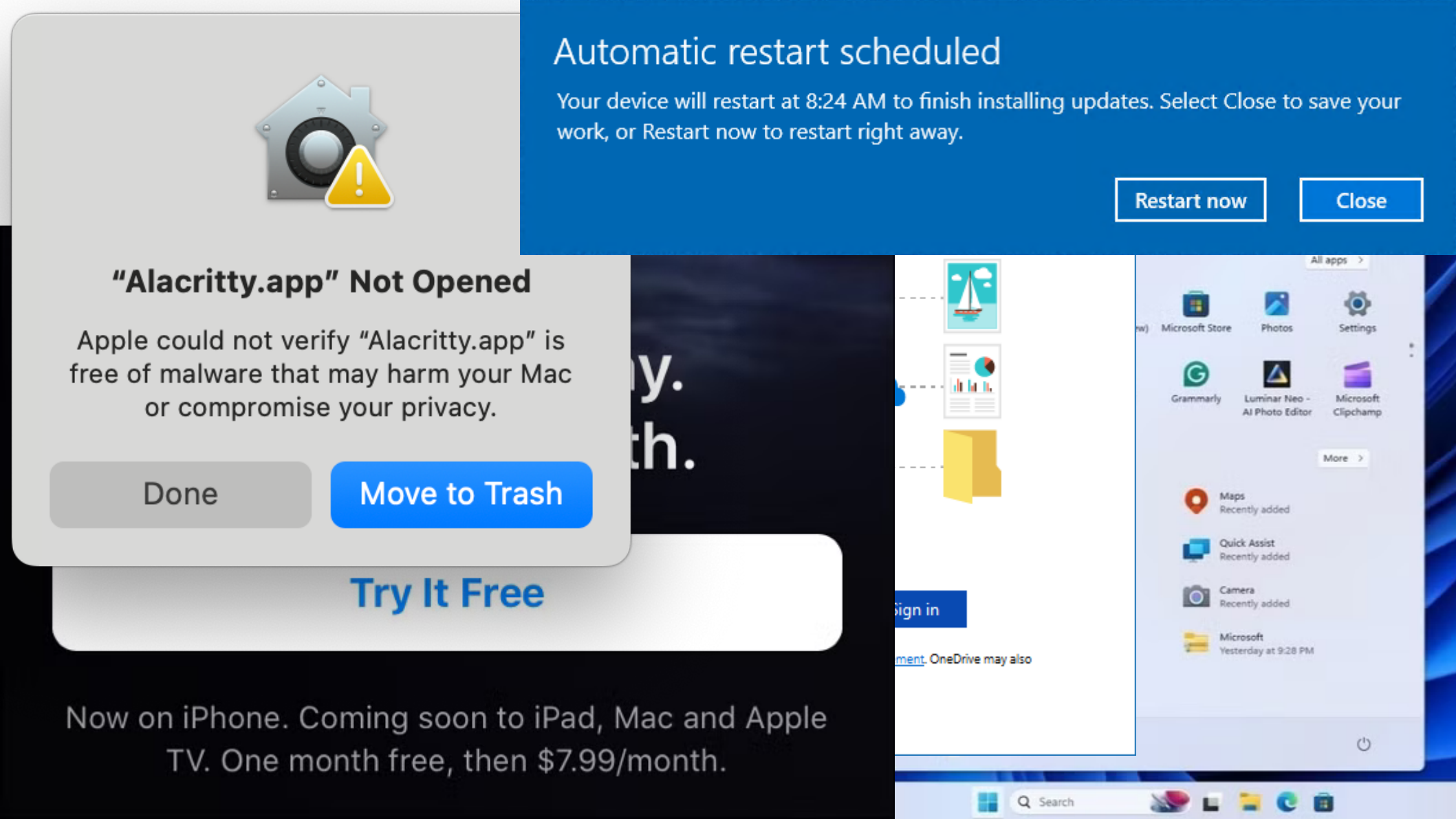




Prop ples


Microsoft OneDrive
Set up OneDrive
Put your files in OneDrive to get them from any device.
Sign in
ment. OneDrive may also
Arcade
All you can play.
Free for 1 month.
Try It Free
Now on iPhone. Coming soon to iPad, Mac and Apple TV. One month free, then $7.99/month.
All apps
Microsoft Store
Photos
Settings
Grammarly
Luminar Neo - AI Photo Editor
Microsoft Clipchamp
More
Maps
Recently added
Quick Assist
Recently added
Camera
Recently added
Microsoft
Yesterday at 9:28 PM
Search


ples
“Alacritty.app” Not Opened
Apple could not verify “Alacritty.app” is free of malware that may harm your Mac or compromise your privacy.
Done
Move to Trash
up OneDrive
OneDrive to get them from any device.
Try It Free
Now on iPhone. Coming soon to iPad, Mac and Apple TV. One month free, then $7.99/month.
Sign in
OneDrive may also
All apps
Microsoft Store
Photos
Settings
Grammarly
Luminar Neo - AI Photo Editor
Microsoft Clipchamp
More
Maps
Recently added
Quick Assist
Recently added
Camera
Recently added
Microsoft
Yesterday at 9:28 PM
Search


Automatic restart scheduled
Your device will restart at 8:24 AM to finish installing updates. Select Close to save your work, or Restart now to restart right away.
Restart now
Close
"Alacritty.app" Not Opened
Apple could not verify "Alacritty.app" is free of malware that may harm your Mac or compromise your privacy.
Done
Move to Trash
Try It Free
Now on iPhone. Coming soon to iPad, Mac and Apple TV. One month free, then $7.99/month.
Microsoft Store
Photos
Settings
Grammarly
Luminar Neo - AI Photo Editor
Microsoft Clipchamp
More
Maps
Recently added
Quick Assist
Recently added
Camera
Recently added
Microsoft
Yesterday at 9:28 PM
Search



Restarting

Open source software examples

- Linux
- Python
- gcc
- vim

Why study proprietary software?

- It is more likely to contain malicious behavior
- But it is more difficult without the source code :)

Summary

- Proprietary software treats people as means to an end
- Open software treats people as ends in themselves

Lab 3: BitBomb

- Cliche of malicious, proprietary software
- Close analog: ransomware

Lab 3: BitBomb

- When you run ./bitbomb you need to enter the correct “defusal” codes
- There are multiple phases and multiple codes you need to enter to defuse it
- How will you know which code to enter? You will have to reverse engineer it!

Lab 3: BitBomb

- We do not have the bitbomb *binary*'s source code
- All we have is its machine code!
- We will have to *disassemble* it – convert from machine code to assembly code
- Then figure out what that assembly is doing!

Lab 3: BitBomb

- Tools for disassembly:
- `objdump -d` 🥲
- `gdb` 👉

scanf

- The inverse of `printf`
- `printf("%d %s\n", some_int, some_str);`
  - Writes formatted output to stdout
- `scanf("%d %s\n", &some_int, &some_str);`
  - Reads formatted input from stdin

sscanf

- The inverse of `sprintf`
- `sprintf(s, "%d %s\n", some_int, some_str);`
  - Writes formatted output to string `s`
- `sscanf(s, "%d %s\n", &some_int, &some_str);`
  - Reads formatted input from string `s`

demobomb

- This one we will have a copy of the source code to “cheat”
- `objdump -d demobomb`
- `gdb demobomb`

disassemble [disas]

- `disas <function>`
  - Example: `disas <main>`
- `disas <address>`
  - Example: `disas 0x401172`

layout asm

- `layout src` – use when we have the source
- `layout asm` – use when we do not

break [b]

- b <function>
  - Example: b main
- b <lineno>
  - Not useful for Lab3 because we don’t have source
- b *<address>
  - Example: b *0x401172

print [p]

- `print $<registername>`
  - Example: `print $rax`
  - Example: `print $eax`
  - Note: $, not % 🥲
- Related: `info registers`
  - Prints all registers

stepi / nexti

- Assembly equivalents of step / next
- Recall:
  - `step [s]` – next line, stepping over functions
  - `next [n]` – next line, stepping into functions
- Above not useful without source! Use:
  - `stepi [si]` – next instruction, stepping over `callq`
  - `nexti [ni]` – next instruction, stepping into `callq`

print [p] / x

- Print string stored at 0x402026:
  - `print (char *) 0x402026`
  - `x/s 0x402026`
- Print int stored at 0x404030:
  - `print (int) *0x404030`
  - `x/wd 0x404030`

set

- Can be used to set (e.g.) register values
- Examples:
  - `set $rax = 0`
  - `set $rip = 0x401172`

finish [f]

- Execute until current function returns
- Accidentally used `nexti` instead of `stepi`?
- Just use `finish`
- Also useful for seeing current function’s return value

call

- `call <funtion call>`
- Does not affect the program counter
- But function side effects can affect memory
- Examples:
  - `call printf("%s %s\n", "hello", "world")`
  - `call (int) foo(20)`
  - Note: if return type is unknown, must be manually specified as cast

Lab 3 summary

- Some of the later phases we need to learn more material
- You know everything you need for the initial phases
- And you might be able to get the later ones too!

TODO

- Assignment 3 assigned due October 28