

Operating System Concepts

EE 5173

王凡

farn@ntu.edu.tw

Dept. of Electrical Engineering
National Taiwan University

目的

- 介紹作業系統的基本概念、組成原理
- 透過實習，培養學生系統程式製作的基礎能力

對象

- 電機系、或外系同學
- 對系統程式、硬體驅動程式，有興趣研究的同學
- 想要瞭解系統程式製作的同學

對學生的課前要求

- 對計算機資料結構的基本認識與能力
- 使用C/C++語言
 - TA 會準備基本的Nachos介紹。

基本資料

- 課號：EE 5173/921 U9580
- 授課教授：王凡
- 上課時間：09:10-12:00, Wednesday；107年秋季
- 教室：博理館 114
- 教科書：Silberschatz, Galvin, and Gagne,
Operating System Concepts
Version 9, John Wiley & Sons, Inc., 2010.
台灣代理：新月圖書公司
<http://www.bookcake.com.tw>
newmoo@ms15.hinet.net

教師聯絡方式

- 課程網址：

<https://ceiba.ntu.edu.tw>

<http://cc.ee.ntu.edu.tw/~farn/courses/OS/index.htm>

- 教師網址：

<http://cc.ee.ntu.edu.tw/~farn>

- email:

farn@ntu.edu.tw

助教資訊

- name: 林其政
- email: a1234567899525@gmail.com
- Office: BL 618
- Office hour: 13:20-14:20

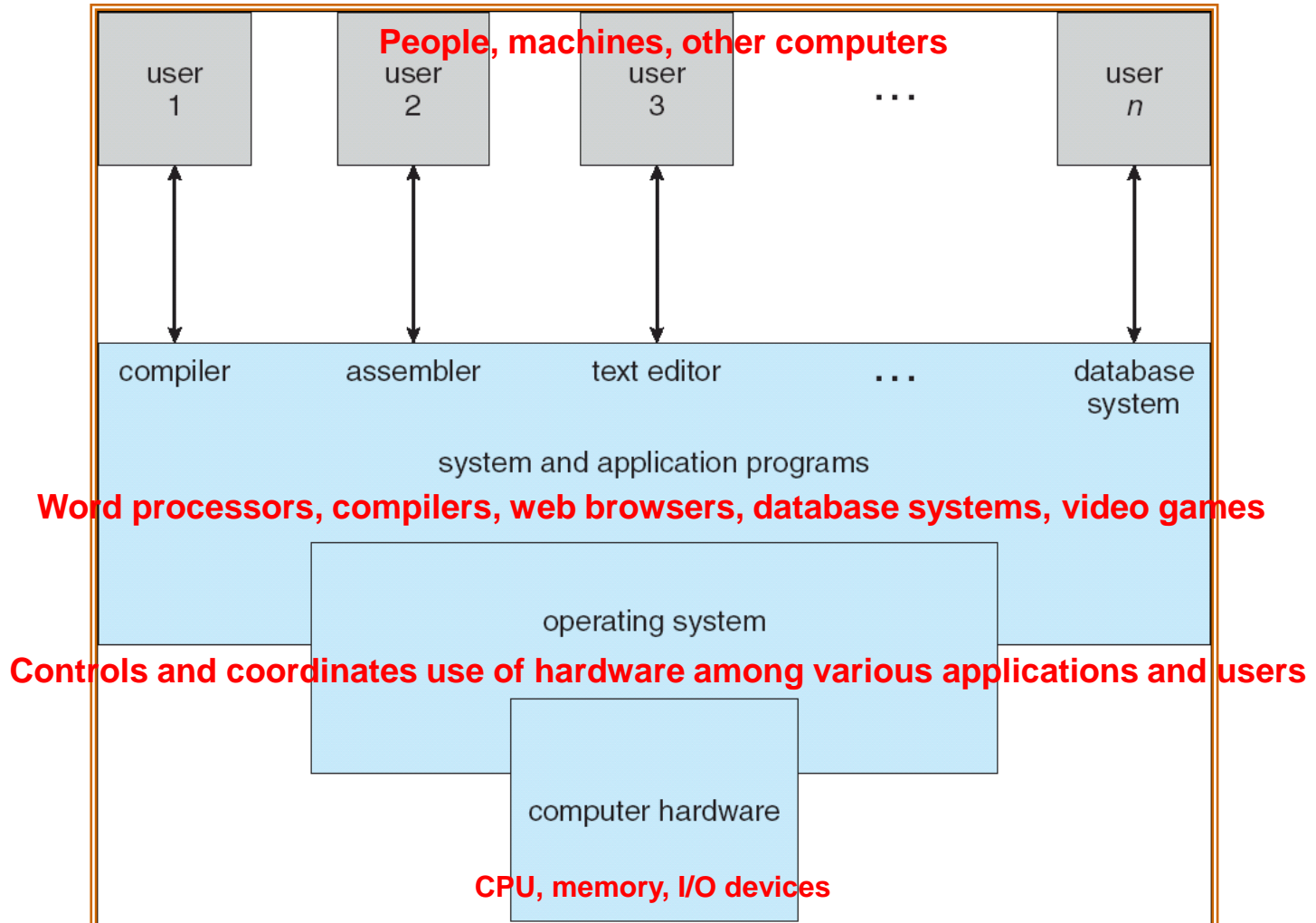
Wednesday



What is an Operating System?

- A program that acts as an intermediary between
 - a user of a computer and
 - the computer hardware.
- Operating system goals:
 - Execute user programs and make solving user problems easier.
 - Make the computer system convenient to use.
 - Use the computer hardware in an efficient manner.
 - Safety and security in the new era.

Four Components of a Computer System



Functions of Operating Systems

- **Oversee operation of computer**
 - bridge between software and central hardware
- **Management of system resources**
- **Store and retrieve files**
- **Schedule programs for execution**
- **Coordinate the execution of programs**
- **Protections of system resources**
- **Security**

Operating System Definition

- OS is a **resource allocator**
 - Manages all resources
 - Decides between conflicting requests for efficient and fair resource use
- OS is a **control program**
 - Controls execution of programs to prevent errors and improper use of the computer

Operating System Definition (Cont.)

- No universally accepted definition
- “Everything a vendor ships when you order an operating system” is good approximation
 - But varies wildly
- “The one program running at all times on the computer” is the **kernel**.
 - Everything else is either a system program (ships with the operating system) or an application program

Why do we want to learn O.S. ?

Learn

- How to program embedded systems
 - Interfaces between hardware and software
- The tremendous collections of software technology
 - Scheduling theory, Distributed computing, Storage hierarchy, Security systems
- Learn **SYSTEM Technology**
 - *Simple cute apps do carry your career.*

Evolution of Modern Computer (1/7)

How it started ?

Before computers:

- Who are we ?
- What is intelligence ?
- What is the mechanism of thinking ?

The secret of human intelligence
*The source of
ultimate curiosity*

Evolution of Modern Computer (2/7)

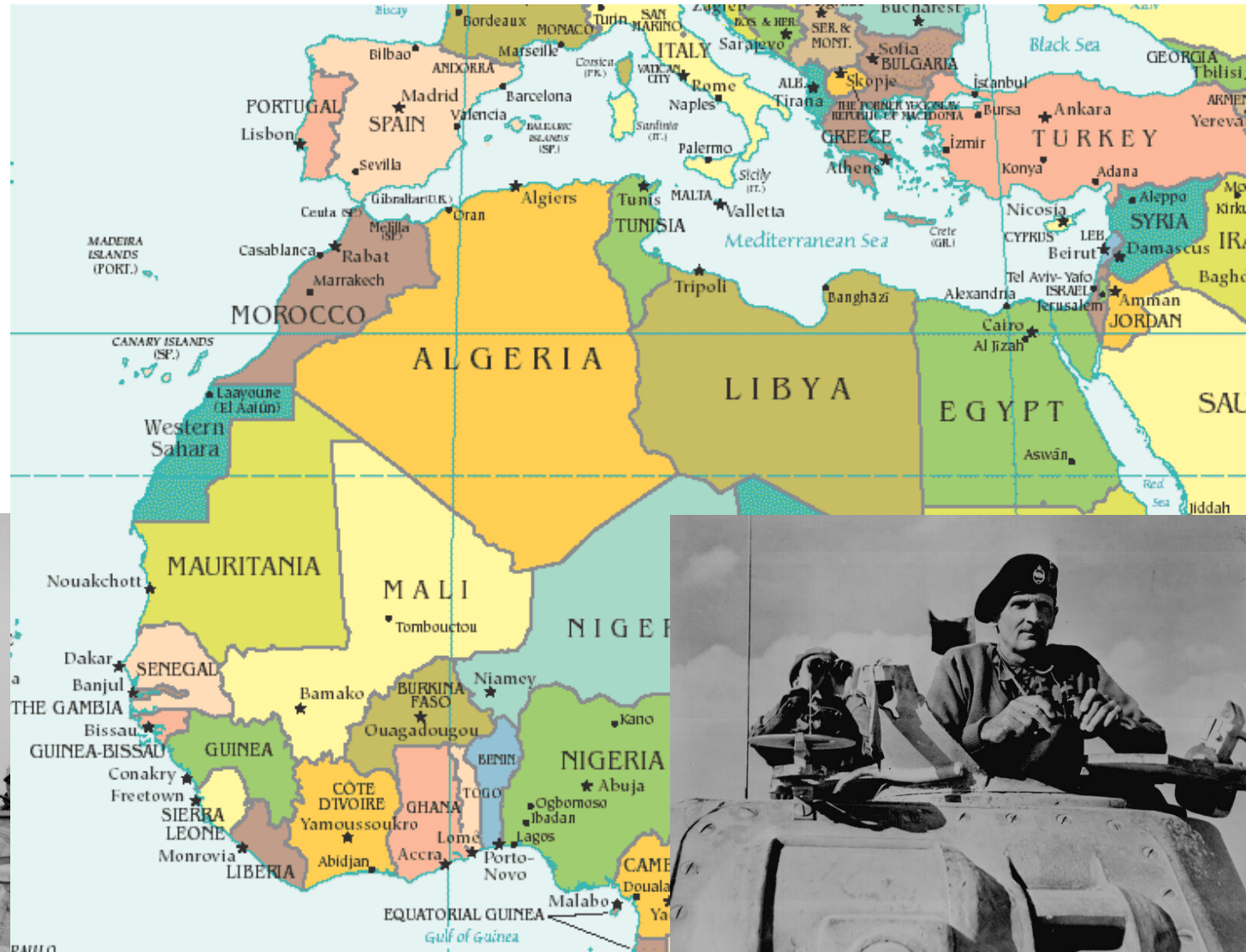
How it started ?

Before computers:

- Abacus, China
- GCD algorithm, Euclid (Greece, B.C.)
- The origin of knowledge, Aristotle (Greece, B.C.)
- Philosophy of Mind (17th century)
- Computing machines (18-19th century)
- Hollerith's machine for

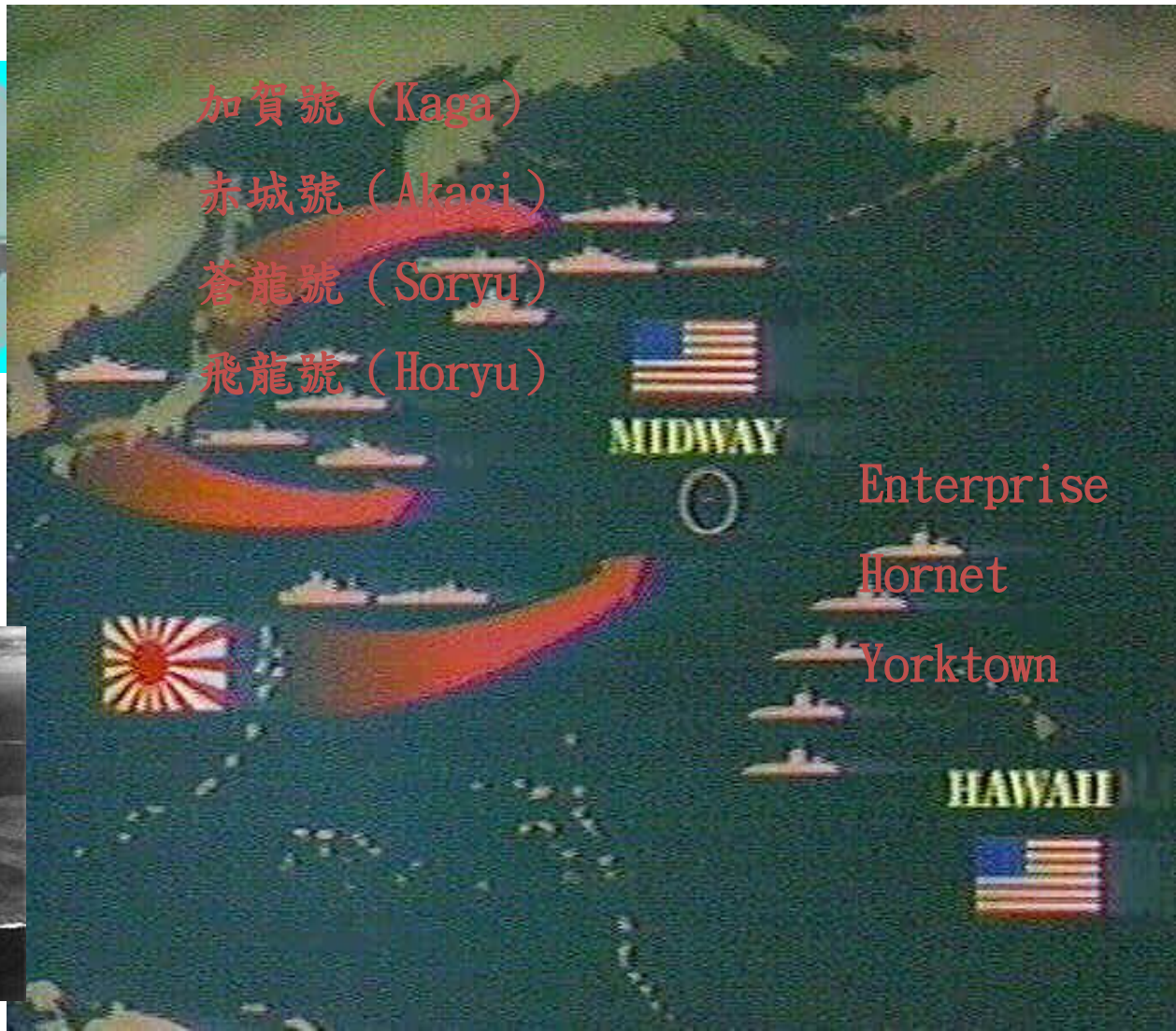
Evolution of Modern Computer (3/7)

The War in North Africa, WWII



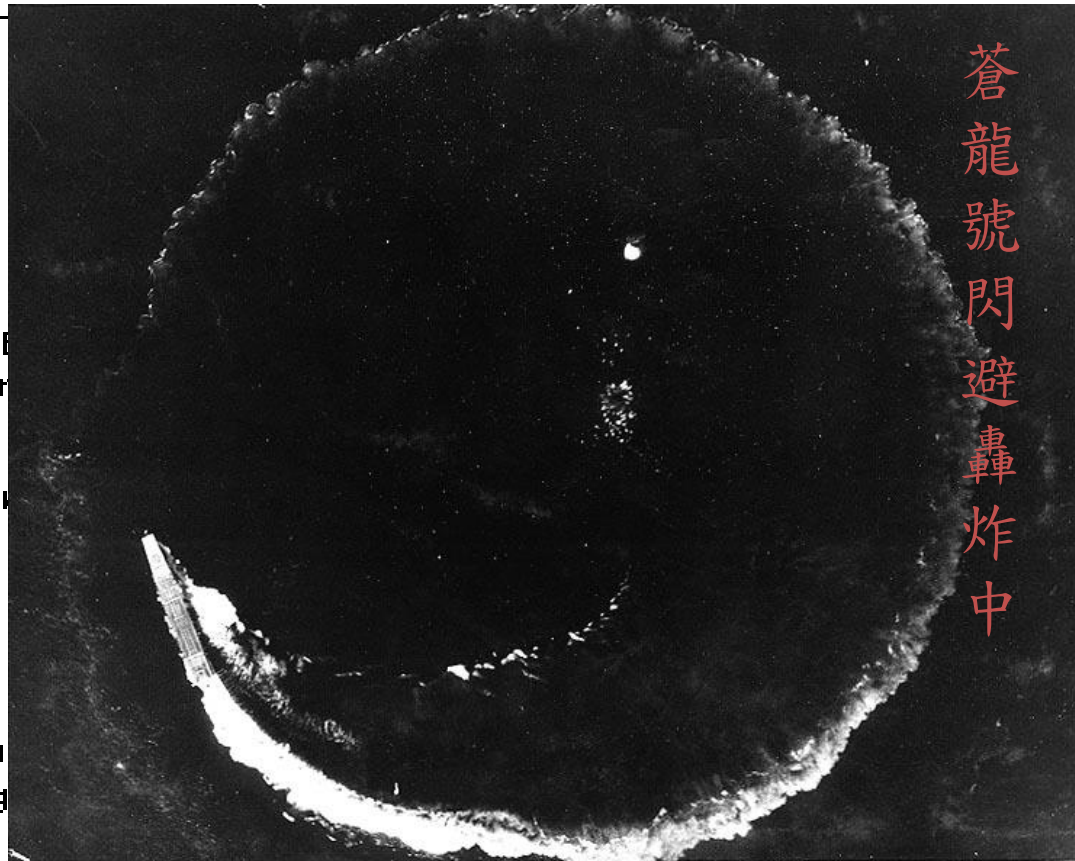
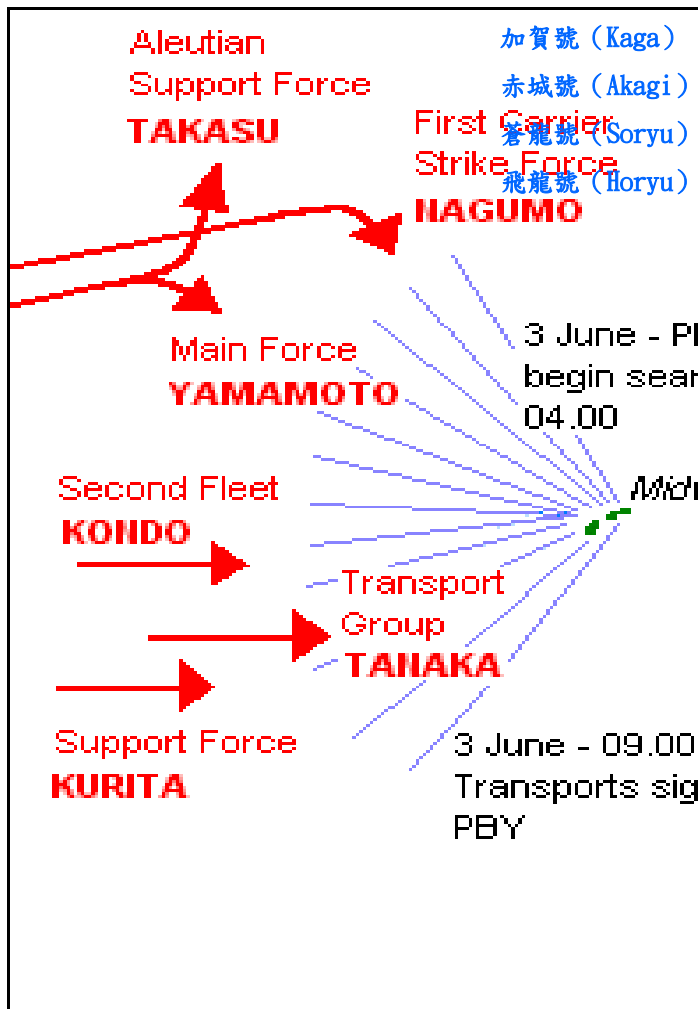
Evolution of Modern Computer (4/7)

Battle of Midway, June 4, 1942



Evolution of Modern Computer (5/7)

Battle of Midway, June 4, 1942



蒼龍號閃避轟炸中

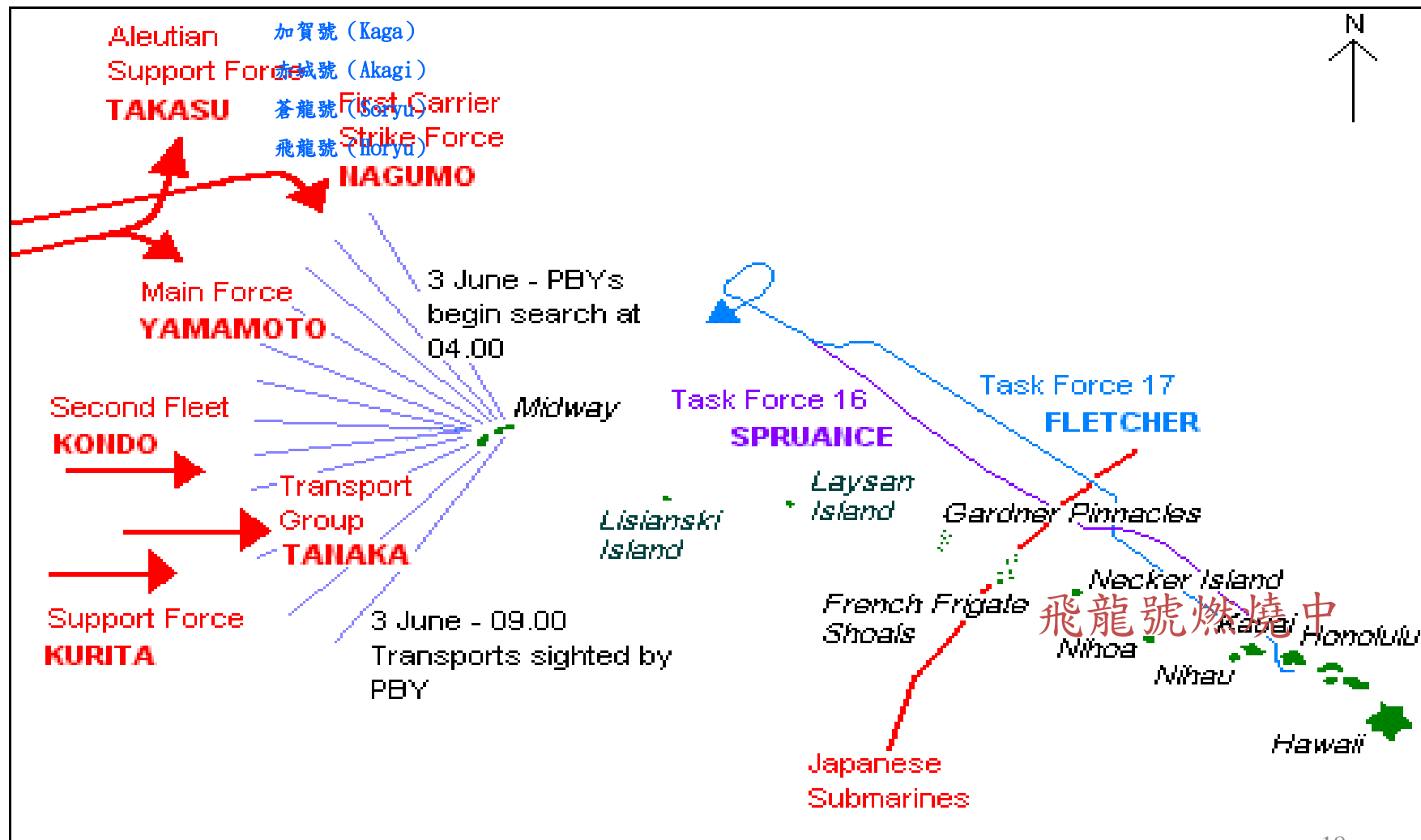
Photo # USAF ID 4845 Soryu under B-17 attack during Battle of Midway

Japanese
Submarines

HAWAII

Evolution of Modern Computer (6/7)

Battle of Midway, June 4, 1942



The Battle for Midway, Map 1: The Fleets Approach Midway

Evolution of Modern Computer (7/7)

The 1st Modern Computer

ENIAC, the first computer

- In University of Pennsylvania
- Only hardware and programs
- The computers ? Men or machines ?

Evolution of Operating Systems (1/6)

- The period of no OS
 - operator = programmer
 - loading programs from console panel switches, cards, paper tapes
 - monitoring through display lights
 - error handling by operators
 - output to tapes or cards
- later → card readers, line printers, magnetic tapes

Evolution of Operating Systems (2/6)

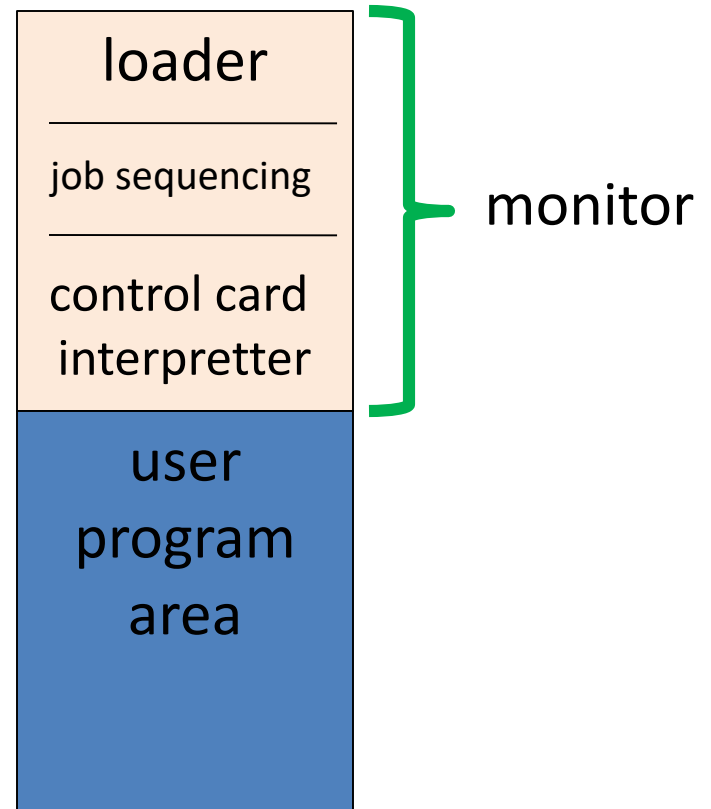
- The later period of no OS
 - card readers, line printers, magnetic tapes
 - assemblers
 - loaders, linkers
 - libraries
 - device drivers
 - high-level languages: FORTRAN, COBOL, ...
 - compilers

Evolution of Operating Systems (3/6)

- Before the birth of OS
 - high-level languages: FORTRAN, COBOL, ...
 - compilers
 - human intervention → low CPU utilization
 - operator ≠ programmer
 - batch processing to share loading & setup time.

Evolution of Operating Systems (4/6)

- The birth of OS
 - Automatic job sequencing
 - resident monitor
 - always staying in the memory



memory allocation
for a resident monitor

Evolution of Operating Systems (5/6)

- Parallel I/O
 - magnetic tapes
 - sequential access
 - offline preparation of program and data to magnetic tapes
 - disks
 - random access
 - evolving to replace magnetic tapes
 - Spooling
 - simultaneous peripheral operation on-line
 - memory → disks → I/O

Evolution of Operating Systems (6/6)

- Shared Computing
 - Batch processing
 - Interactive processing
 - Requires real-time processing
 - Time-sharing/Multitasking
 - Implemented by Multiprogramming
 - Multiprocessor machines

成績評量：(subject to changes)

- 期中論文心得報告(30%)
 - a regular paper published in **ACM SOSP 2018, ACM ISCA 2018, IEEE ICPP 2018, ACM POPL 2018, or ACM PODC 2018.**
- 期末考(30%)、
- 作業(10%)、
 - 作業題目，選自課本習題，公佈於課程網頁。
繳交期限，為該章授課結束後一週。
- 學期Nachos計畫 (30%)

成績評量：(subject to changes)

- 期中論文心得報告(30%)、
 - 各自選擇一個chapter的技術主題，撰寫與Google、Facebook、Microsoft、Apple等軟體業的技術關連。
 - 9/26前，將主題email給助教，
 - 10/3前與助教確定主題。
 - 10/24前email三頁或以上之WORD報告，給助教。

學期計畫

- Nachos作業系統實習。
- 在任何支援Linux的環境，都可執行。
- 模擬MIPS硬體架構。
- 一人一組，每人報告5mins一次。
 - 請儘速向助教登記你要報告哪一個計畫，先填先選。
- 目前暫訂三個部分：
 - Thread management
 - CPU scheduling
 - Virtual memory management

學期上台報告

- 每位同學需上台報告一次。
- 可以是下列四類之一。
 - 學期論文心得報告（限40人）
 - 第一次學期計畫報告（限15人）
 - 第二次學期計畫報告（限15人）
 - 第三次學期計畫報告（限15人）

注意：

- 各類報告、計畫，每位同學都要做。但上台報告擇一。
- 每次報告以6分鐘為限。

預定課程進度 (I)

1. 9/12 課程簡介
2. 9/19 第一單元 (一、二章)
3. 9/26 第一單元 (一、二章)
email期中報告題目、報告順序給助教
繳交學期計畫報告順序
5. 10/3 第一單元、確認期中報告題目與順序
學期計畫介紹：Nachos簡介
第一階段學期計畫介紹：
thread management
6. 10/17 第二單元 (三、四、五章)

預定課程進度 (II)

- 7. 10/24 第二單元、繳交期中報告
- 8. 10/31 第二單元
- 9. 11/7 第三單元 (含六、七章)
第一階段學期計畫報告
第二階段學期計畫介紹：*CPU scheduling*
- 10. 11/14 第三單元
- 11. 11/21 第三單元
- 12. 11/28 第四單元 (八、九章)

預定課程進度 (III)

- 13. 12/5 第四單元 (八、九章)
- 14. 12/12 第四單元、
第二階段學期計畫報告、
第三階段學期計畫介紹：Virtual Memory
- 15. 12/19 第四單元
- 16. 12/26 第五單元 (含10、11、12章)
- 17. 1/2 第五單元
- 18. 1/9 期末考
- 19. 1/16 第三階段學期計畫報告