

## Resources and Office Hours

### Course web page:

<http://www.student.cs.uwaterloo.ca/~cs350/>

### Course newsgroup:

uw.cs.cs350

### IAs (aka tutors):

cs350 at student.cs.uwaterloo.ca

### Me:

sbuettch at plg.uwaterloo.ca

Office hour: Wed, 3:00 – 4:00 pm (DC3548)

# Course Objectives

## Objectives:

- To learn about the facilities provided by an operating system and to understand how they are implemented.
- To learn how to work with a large software project (NachOS has around 15 kLOC) and to write large amounts of code – as a team.
- To learn C and C++.

## How to get there:

- Attend lectures (covers part 1).
- Do the NachOS assignments (covers parts 1, 2, and 3).

# Course Outline

- processes and threads
- concurrency and synchronization
- address spaces and virtual memory
- process/thread scheduling (when to run which process)
- devices and device management (in particular: hard disks)
- file systems (e.g., POSIX, indexing, journalling)
- interprocess communication (sockets, msg queues, shared mem)
- networking (IP, TCP, UDP)
- security (access control, authentication, buffer overflows)

# Grading Scheme

- 3 assignments
- 1 midterm exam
- 1 final exam
- a1, a2, a3: your grades on assignments 1-3 (as percentages)
- m: your midterm grade (as a percentage)
- f: your final exam grade (as a percentage)

```
double getFinalGrade(double a1, double a2, double a3, double m, double f) {  
    double normal = 0.15 * a1 + 0.15 * a2 + 0.1 * a3 + 0.2 * m + 0.4 * f;  
    double exam = (0.2 * m + 0.4 * f) / 0.6;  
    if ((exam < 50) && (exam < normal))  
        return exam;  
    else  
        return normal;  
}
```

# Assignments

## **Assignment 0: Setting up groups**

Form groups of 3, following the procedure described on the course web page.

Due date: Tue, Sep 19 (noon)

## **Assignment 1: Processes and threads; synchronization**

Due date: Tue, Oct 10 (noon) – worth 15% of your final grade

## **Assignment 2: Virtual memory, paging, TLB**

Due date: Mon, Nov 6 (noon) – worth 15% of your final grade

## **Assignment 3: File systems**

Due date: Mon, Dec 4 (noon) – worth 10% of your final grade

## Slip days:

- Slip days can be used to defer the submission of an assignment by 1, 2, or 3 days.
- Each slip day buys you an additional 24 hours after the assignment deadline.
- Each group starts with a total of 6 slip days.
- The last submission of the last assignment can only be deferred by 36 hours (= 2 slip days), because of end of lectures.

# Academic Honesty

## A few words about plagiarism and cheating:

- No cooperation with students that are not in the same group.
- No code or testcases from students that are not in your group.
- No re-use of code submitted in a previous term.

We will check assignment submissions for similarity with other submissions. If you are collaborating with other groups, chances are we will notice it.

**UW is very strict on this. *So, please do not cheat!***

As a side note: If you have a question about an assignment detail, and the question indicates what your solution to the assignment problem is, please do not ask the question on the newsgroup. Send an email instead.

# What is an Operating System?

According to Wikipedia:

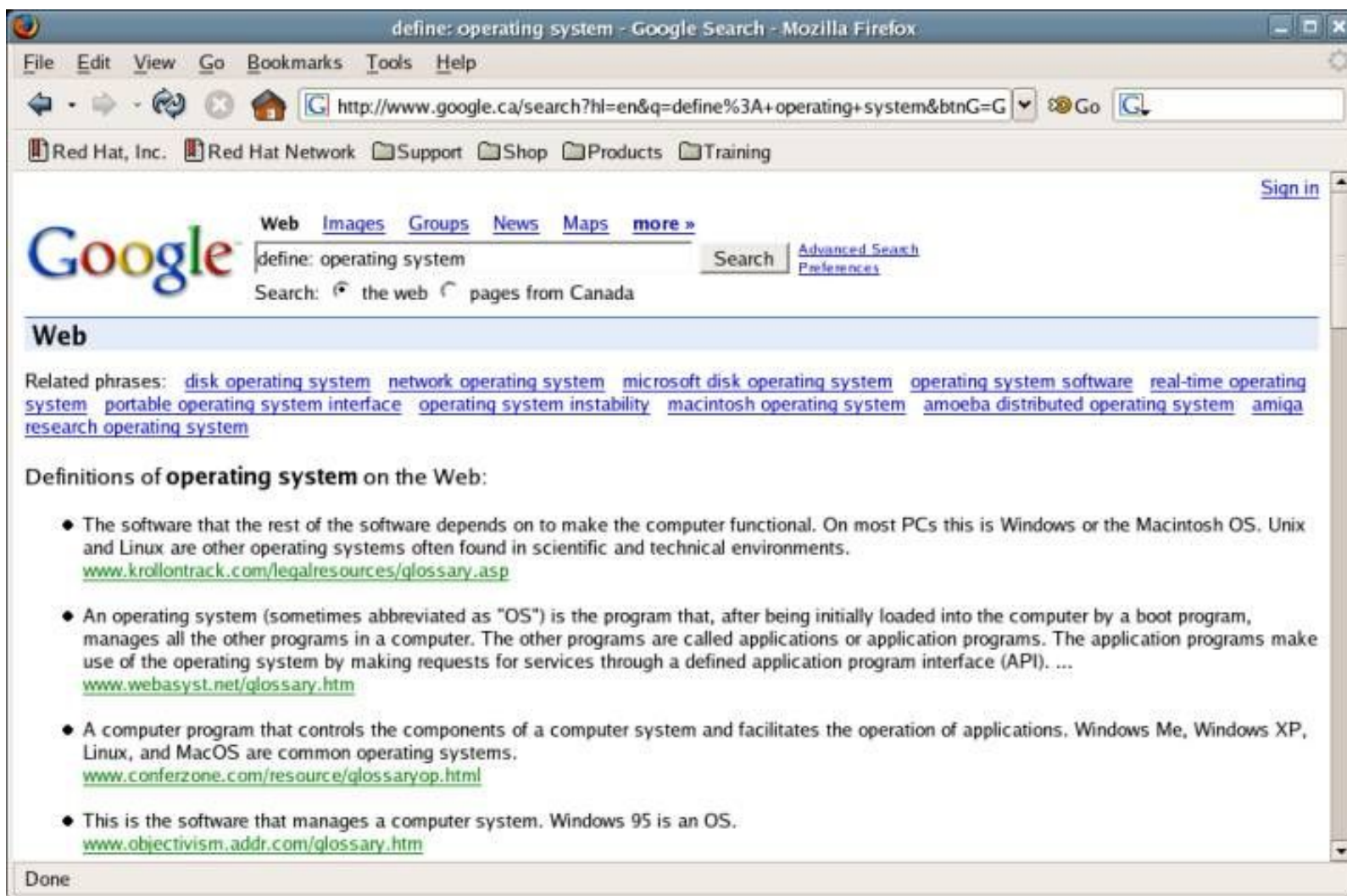
*An operating system (OS) is a software program that manages the hardware and software resources of a computer. A key component of system software, the OS performs basic tasks, such as controlling and allocating memory, prioritizing the processing of instructions, controlling input and output devices, facilitating networking, and managing files.*

([http://en.wikipedia.org/wiki/Operating\\_System](http://en.wikipedia.org/wiki/Operating_System))



# What is an Operating System?

According to Google:



# What is an Operating System?

According to me:

*The operating system is your best friend (and your worst enemy).*

## **The application view of an operating system:**

- It provides memory and processor time to the application.
- It provides an abundance of basic facilities that are used by many programs over and over again: File systems, networking, hardware access.
- It protects a program from other programs: Memory protection and process isolation.
- It protects users from other users: File access permissions.
- If you don't play by its rules, it kills you: Segmentation violation.

# What is an Operating System?

Three views of an operating system:

- **Application view:** What services does it provide?

*Rules for survival, APIs (application programming interfaces), utility programs (ls, cp, etc.).*

- **System view:** What problems does it solve?

*Resource allocation, process scheduling, hardware access.*

- **Implementation view:** How is it built?

*How does it support concurrency? Is it monolithic (Linux) or modular (Minix)? How does it schedule processes? How does it interact with the hardware?*

# Schematic View of an OS



# Operating System Abstractions

## **An operating system is also a set of abstractions:**

- address spaces: abstract view of primary memory (RAM)
- files and directories: abstract view of secondary storage (HDD)
- processes & threads: abstract view of program execution
- sockets: abstract view of interprocess communication

The exact implementation and the underlying details do not matter, as long as these components follow an abstract specification.

## **In this course:**

- Why are the abstractions designed the way they are?
- How are they manipulated by a user program?
- How are they implemented by the OS?