

CSI110

Lecture 1

Introduction to UNIX

Introduction

- Who am I?
- If you haven't already got an account to use the Computer Science and AI UNIX Lab, please apply for one at the Computer Services Centre

- Web site for CSI110...

<http://www.cs.um.edu.mt/~cstaff/courses/lectures/csi110/index.html>

- Recommended reading:

The Web site for the course

Mark G. Sobell. *UNIX System V - A Practical Guide*. Benjamin Cummings, ISBN: 080537566X

© 2001. Christopher Staff. Department of Computer Science and AI, University of Malta.

CSI110: Lecture 1

1

Departmental Resources

- The Department of Computer Science and AI uses a wide range of computing platforms (UNIX, Windows, and MacOS), and a number of courses require you to be familiar with UNIX
- The UNIX computer you will be using is babe (babe.cs.um.edu.mt)
- You can access babe from either of the Department's computer labs, or from any other computer in the world, via the Internet
- If you access babe from the Dept's UNIX lab, then you will have a Graphical User Interface (the Common Desktop Environment) available
- From any other computer (including those in the NT lab) you will probably only have telnet access to a character-based (text-only) user interface

© 2001. Christopher Staff. Department of Computer Science and AI, University of Malta.

CSI110: Lecture 1

3

© 2001. Christopher Staff. Department of Computer Science and AI, University of Malta.

CSI110: Lecture 1

2

- UNIX is one of the most popular operating systems in use
- Highly prevalent in industry and academia
- Less common for home use, although free UNIX versions are available (e.g., Linux)

© 2001. Christopher Staff. Department of Computer Science and AI, University of Malta.

CSI110: Lecture 1

4

So, what's an Operating System?

Types of Operating Systems

- Single process, single user
E.g., MS-DOS, CP/M, pre-Windows 95
- Multiple process, single user
E.g., MacOS, Windows 95, 98, 2000, NT
- Multiple process, multi-user
E.g., UNIX, VMS (DEC), VM (IBM)

© 2001. Christopher Staff. Department of Computer Science and AI, University of Malta.

CSI110: Lecture 1

5

Multitasking

- If an OS is capable of executing multiple processes simultaneously, then is called a *multitasking operating system*
- However, if the computer has only one processor, then it is still only capable of executing one process at a time
- A multitasking operating system can switch between different processes, giving each a certain amount of CPU time before switching to another process
- A computer which has n processors, where $n > 1$, (a *multiprocessor*) can execute n processes simultaneously - but still only one per processor
- A multitasking operating system must *protect* processes in memory
- A multiuser operating system must also protect users from each other, and the operating system from user processes

© 2001. Christopher Staff. Department of Computer Science and AI, University of Malta.

CSI110: Lecture 1

7

© 2001. Christopher Staff. Department of Computer Science and AI, University of Malta.

CSI110: Lecture 1

6

- A multitasking operating system must *schedule* processes, so that they have a fair chance of acquiring the *resources* that they need to execute
- So a multitasking operating system must also provide *resource management* facilities
- Scheduling, resource and process management, and the architecture of operating systems is covered in CSM202

© 2001. Christopher Staff. Department of Computer Science and AI, University of Malta.

CSI110: Lecture 1

8

Types of processes (in a multitasking OS)

- Interactive
- Batch
- Server processes or *daemons*
- UNIX supports all three types of processes

So, what is UNIX?

- UNIX is a multiprocess, multi-user OS
- UNIX does not support GUIs
- However, in the UNIX lab, the computers are installed with X- Windows (a application which displays and manipulates GUI screen elements), and the Common Desktop Environment (CDE), an X application
- UNIX has three important features:

Kernel: The actual OS - interface to hardware and software

Shell: Interface between the user and the OS (kernel). Provides a *command interpreter* to accept, process, and respond to user input and *shell scripts*

File System: long-term storage, managed by the OS

The Structure of UNIX

The Shell

An application program which takes input from the user and which interfaces with the OS to execute the input

- The shell *parses* the whole command line, to perform filename expansion, to locate the program that the user has requested, and to ensure that the user has the correct permissions to execute the program

The system call interface

- The system call interface defines UNIX, and different *flavours* of UNIX can implement different system calls
- This leads to incompatibilities between different UNIXes, and portability issues
- POSIX is a system call interface standard. Most flavours of UNIX and

some non-UNIX OSes (e.g., Microsoft Windows NT) are POSIX-compliant

The kernel

- The actual OS
- It responds to system calls, and controls all operations in the computer
- The kernel contains low-level (primitive) functions for:
 - file system management
 - input/output (access to devices)
 - memory management
 - process scheduling
- When a user starts a program, the shell places it into the list of running programs - after that, the kernel is responsible for its execution during its lifespan

The hardware

- Not part of UNIX!
- The kernel is re-written for specific hardware platforms, but the upper layers are unaware of the underlying hardware