Chapter 9: Main Memory Operating System Concepts – 10th Edition Silberschatz, Galvin and Gagne (C)2018, revised by S. Weiss 2020 Chapter 9: Memory Management ?the instructions to loading the base and limit registers are privileged Operating System Concepts – 10th Edition 9.7 Silberschatz, Galvin and Gagne (C)2018, revised by S. Weiss 2020 Address Binding ?Example: The Intel 32 and 64–bit Architectures Operating System Concepts – 10th Edition 9.2 Silberschatz, Galvin and Gagne (C)2018, revised by S. Weiss 2020 Objectives ?We can provide this protection by using a pair of base and limit registers define the logical address space of a process Operating System Concepts – 10th Edition 9.6 Silberschatz, Galvin and Gagne (C)2018, revised by S. Weiss 2020 Hardware Address Protection ?To discuss various memory–management techniques, Operating System Concepts – 10th Edition 9.3 Silberschatz, Galvin and Gagne (C)2018, revised by S. Weiss 2020 The Model ?Protection of memory required to ensure correct operation Operating System Concepts – 10th Edition 9.5 Silberschatz, Galvin and Gagne (C)2018, revised by S. Weiss 2020 Protection ?Operating System Concepts – 10th Edition 9.4 Silberschatz, Galvin and Gagne (C)2018, revised by S. Weiss 2020 Background ?Programs on disk, ready to be brought into memory to execute form an input queue ?CPU must check every memory access generated in user mode to be sure it is between base and limit for that user ?Compiled code addresses bind to relocatable addresses ?A running process generates a stream of memory references : ?machine code fetches instructions, data, and stores data, so we can view it as a memory reference generator.Executable programs are loaded into memory from disk ?Cache sits between main memory and CPU registers ?Addresses represented in different ways at different stages of a program's life ?Contiguous Memory Allocation ?To understand how memory is managed at both the hardware level and the operating system level ?Main memory and registers are the only storage CPU can access directly ?addresses + read requests, or ?Register access is done in one CPU clock (or less) ?Main memory can take many cycles, causing a stall ?Need to censure that a process can access only access those addresses in its address space.Inconvenient to have first user process physical address always at 0000 ?Source code addresses usually symbolic ?Paging ?Swapping ?We use this abstraction to understand how memory is managed.Memory unit only sees a stream of: ?address + data and write requests ?Background ?Structure of the Page Table ???Without support, ??must be loaded into address 0000 ?How can it not be