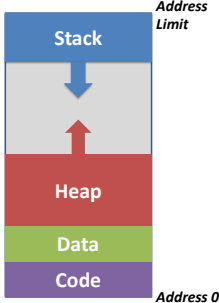# Demand Paging

Dr Andrew Scott

a.scott@lancaster.ac.uk

1

Growing Regions

- As stack or heap expand
  - More pages added as new memory frames referenced in code

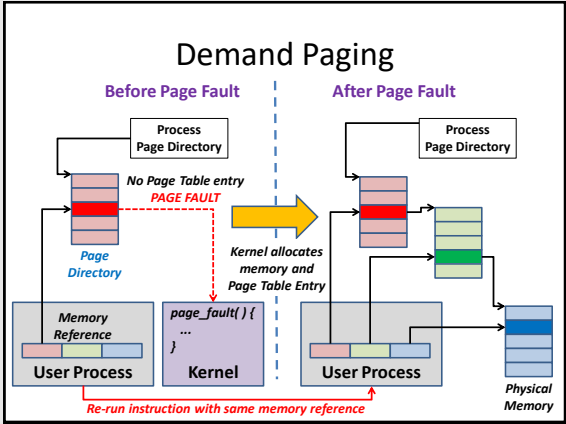- Easily handled by handling Page Faults and *Demand Paging*

Address Limit

Stack

Heap

Data

Code

Address 0

2

Demand Paging

- Start process with partial or even empty page tables

- Reference to unallocated memory will cause a PAGE FAULT

1. PAGE FAULT handler
   - If necessary, creates new Page Table and attaches it to Page Directory

   - Allocates physical memory for frame (memory block), and retrieves code or data from disk

2. Re-run last instruction, which should now succeed

3

Demand Paging

Before Page Fault | **After Page Fault**

Process Page Directory

No Page Table entry
PAGE FAULT

Page Directory

Kernel allocates memory and Page Table Entry

Memory Reference

page_fault( ) {
...
}

User Process | **Kernel**

Re-run instruction with same memory reference

Process Page Directory

User Process

Physical Memory

4