Process Swapping in UNIX V6

Lions' commentary chapter 14 in detail

Urs Fässler

Systems Group
ETH Zürich

20.04.2010

why writing RAM to a disc?

- small device with 16 MB RAM and a Hard disc

WL-HDD (www.asus.com)

why writing RAM to a disc?

- small device with 16 MB RAM and a Hard disc
- want organise thousands of Mp3 on it

WL-HDD (www.asus.com)

OpenJukebox (openjukebox.origo.ethz.ch)

why writing RAM to a disc?

- small device with 16 MB RAM and a Hard disc
- want organise thousands of Mp3 on it

! database doesn't fit into RAM

WL-HDD (www.asus.com)

OpenJukebox (openjukebox.origo.ethz.ch)

why writing RAM to a disc?

- small device with 16 MB RAM and a Hard disc
- want organise thousands of Mp3 on it

! database doesn't fit into RAM

$\Rightarrow$ use virtual memory

WL-HDD (www.asus.com)

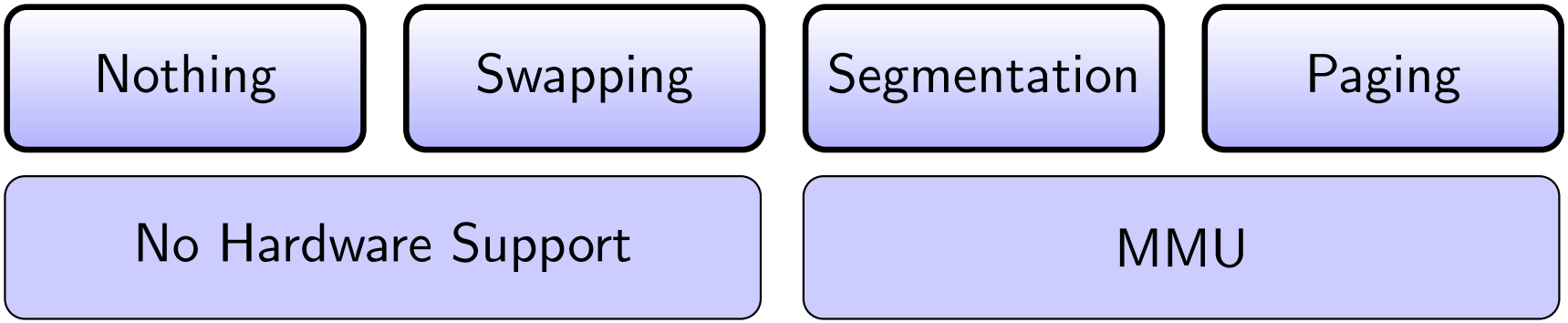
OpenJukebox (openjukebox.origo.ethz.ch)

types of virtual memory

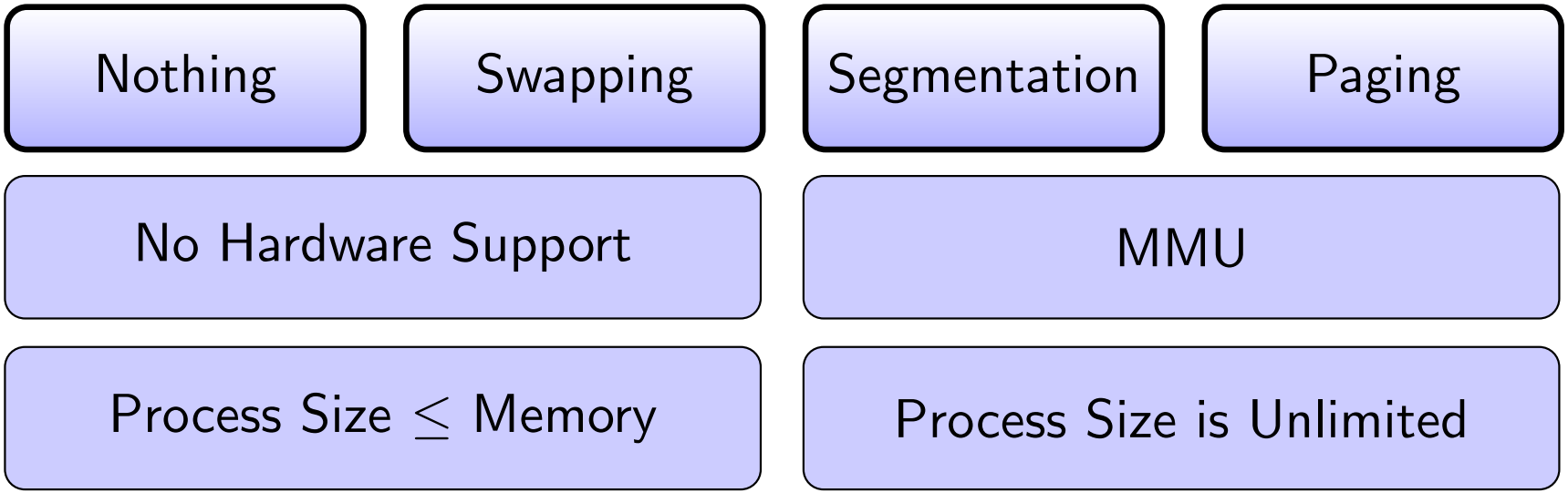
types of virtual memory

Nothing | Swapping | Segmentation | Paging

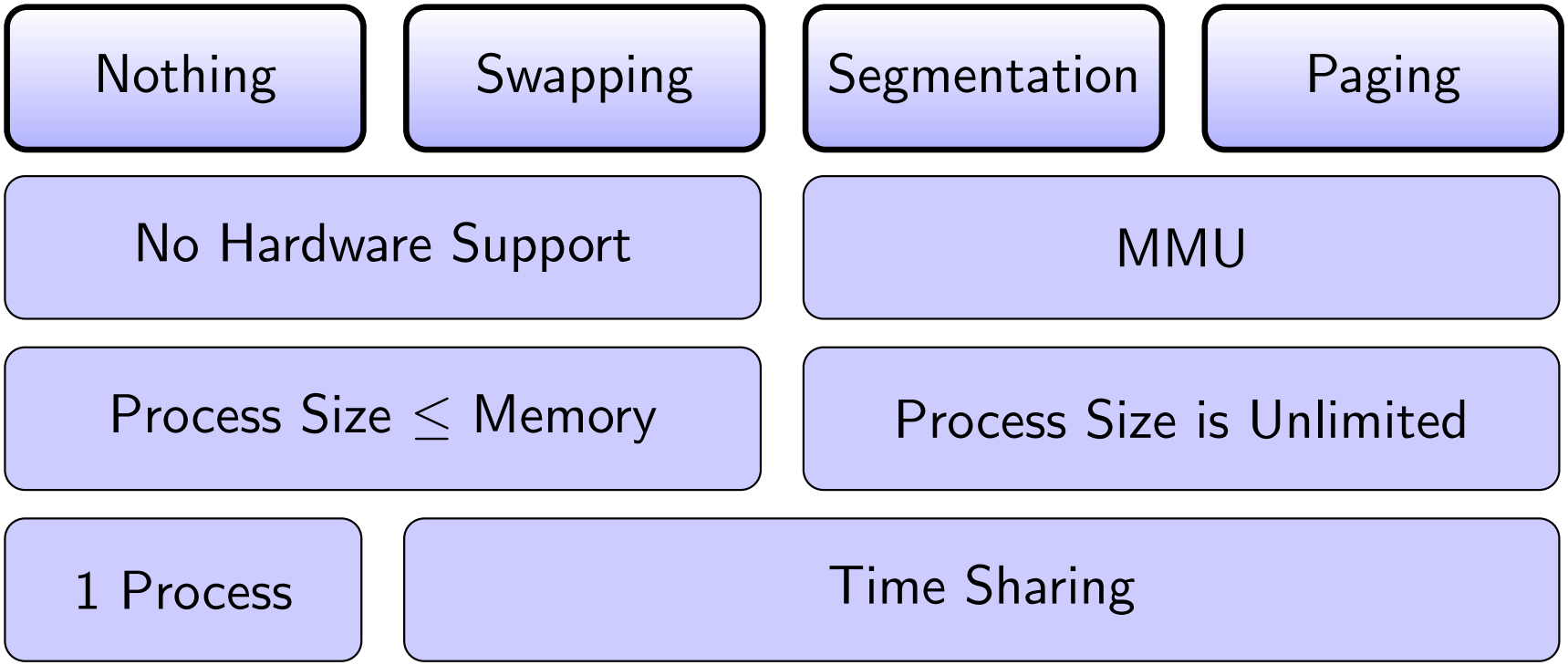
types of virtual memory



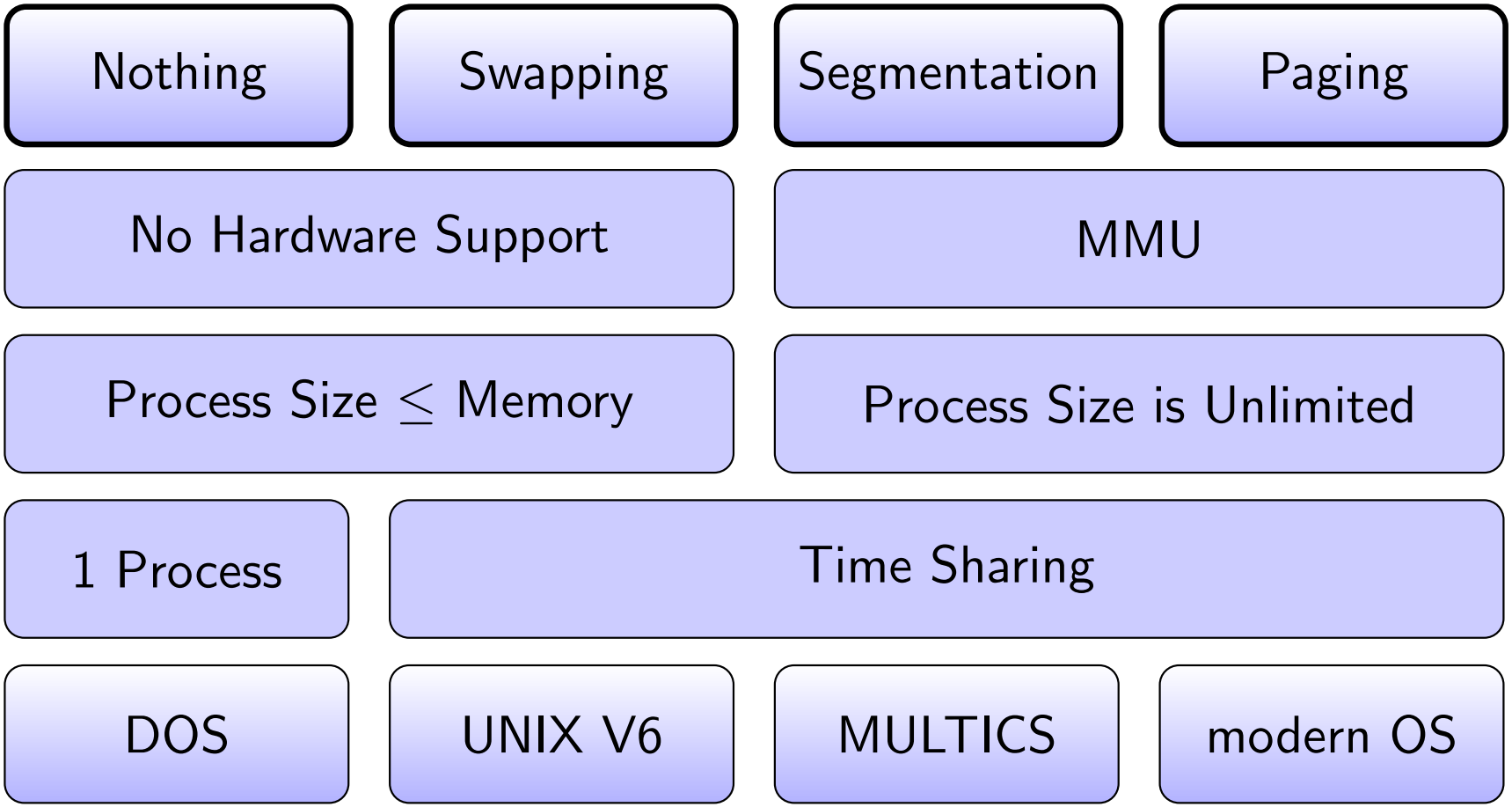
types of virtual memory



types of virtual memory



types of virtual memory



swapping versus paging

swapping	**paging**
older	modern
easy	complicated

swapping versus paging

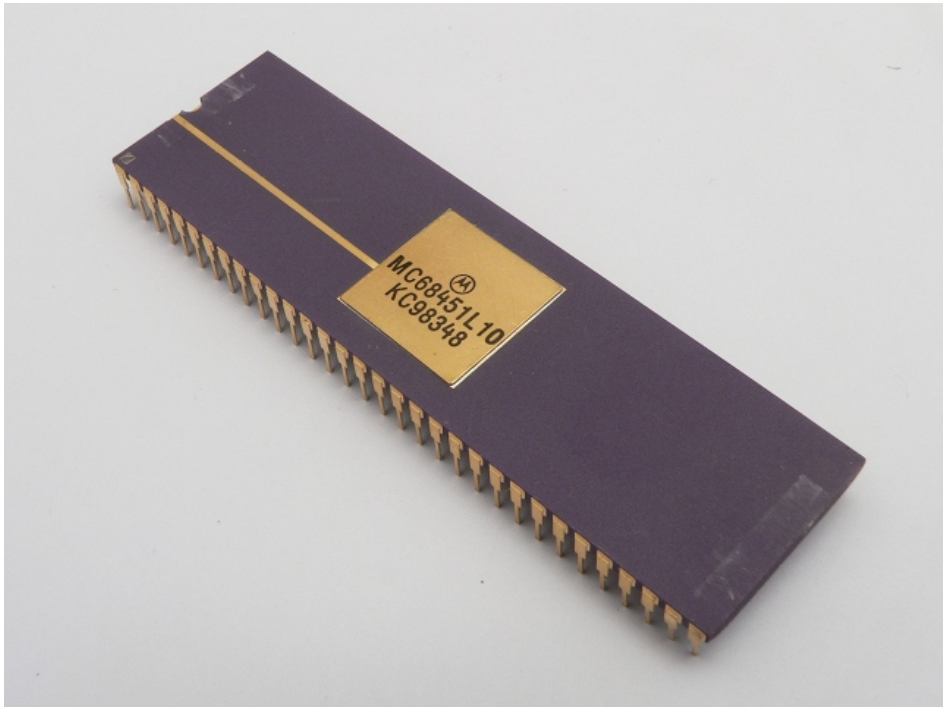
swapping	**paging**
older	modern
easy	complicated
whole process	parts of the memory
proc size $<$ RAM	arbitrary proc size

swapping versus paging

swapping	**paging**
older	modern
easy	complicated
whole process	parts of the memory
proc size $<$ RAM	arbitrary proc size
no special hardware	MMU

MMU - Memory Management Unit

- paging needs a MMU
- integrated in modern CPU such as ARM, IA-32, MIPS
- basic operation of paging is defined by the MMU



www.wikipedia.org

OS tasks in a paging system

- creation of a process
  - create pages
  - copy text and data into pages

OS tasks in a paging system

- creation of a process
  - create pages
  - copy text and data into pages
- set a process running
  - initializing MMU
  - switch actual page to new process

OS tasks in a paging system

- creation of a process
  - create pages
  - copy text and data into pages
- set a process running
  - initializing MMU
  - switch actual page to new process
- page fault
  - find page on swap
  - find free frame in RAM
  - copy page into frame
  - restart last instruction

OS tasks in a paging system

- creation of a process
  - create pages
  - copy text and data into pages
- set a process running
  - initializing MMU
  - switch actual page to new process
- page fault
  - find page on swap
  - find free frame in RAM
  - copy page into frame
  - restart last instruction
- after termination of a process
  - free frames in RAM
  - free pages on swap

OS tasks in a swapping system (UNIX V6)

- creation of a process
  - swap new process out if it doesn't fit in RAM

OS tasks in a swapping system (UNIX V6)

- creation of a process
  - swap new process out if it doesn't fit in RAM
- process size is increased
  - swap process out if it don't fit in RAM anymore

OS tasks in a swapping system (UNIX V6)

- creation of a process
  - swap new process out if it doesn't fit in RAM
- process size is increased
  - swap process out if it don't fit in RAM anymore
- swapper is triggered to do something
  - swap oldest process in (if enough RAM is available)
  - otherwise swap an inactive process out

study of the code

- original code is not very readable

study of the code

- original code is not very readable
- automated formatted with bcpp

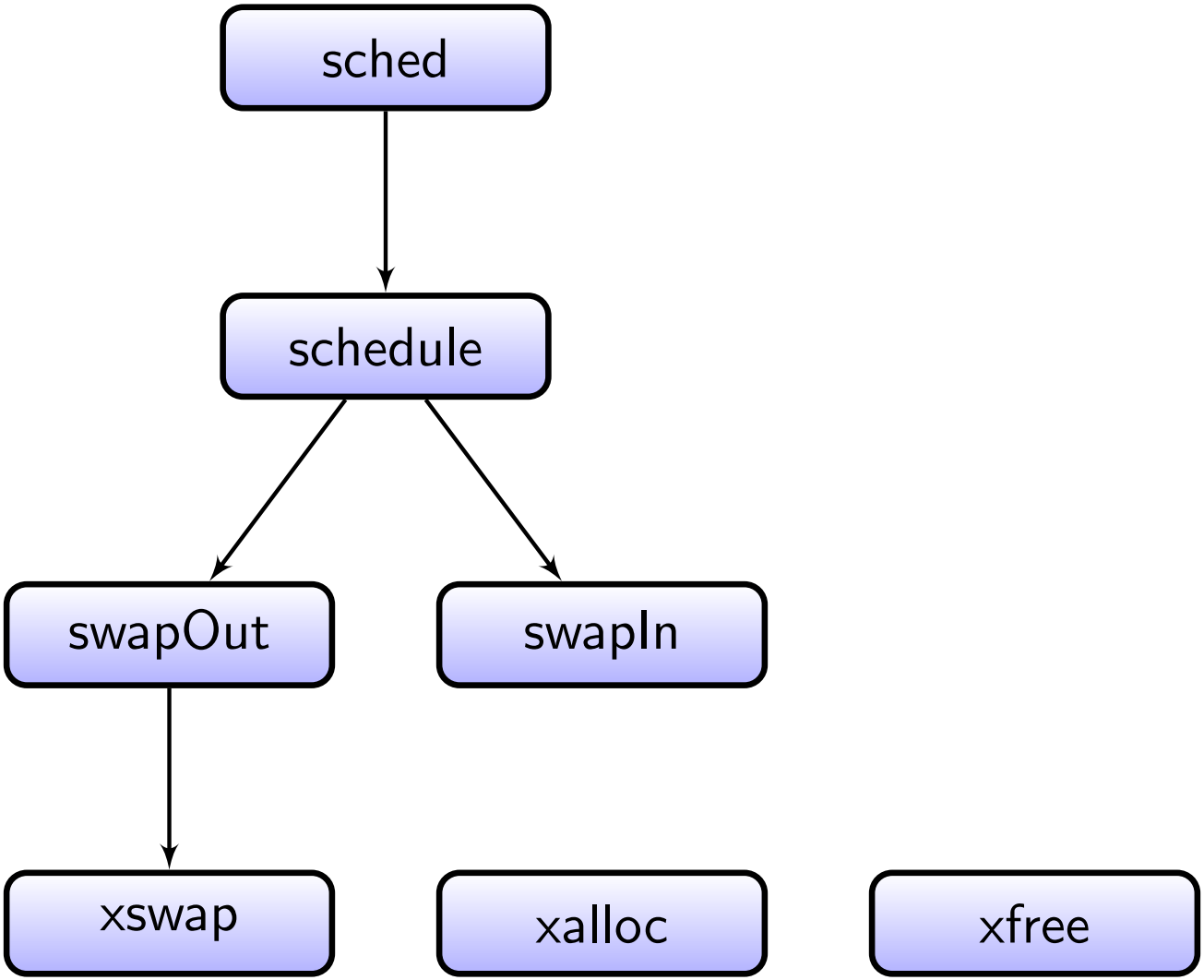
study of the code

- original code is not very readable
- automated formatted with bcpp
- refactored by hand to improve code metrics
  - # of goto: from 11 to 0
  - split sched (100 lines) into 9 functions (and moved to sched.c)
  - used strong typing

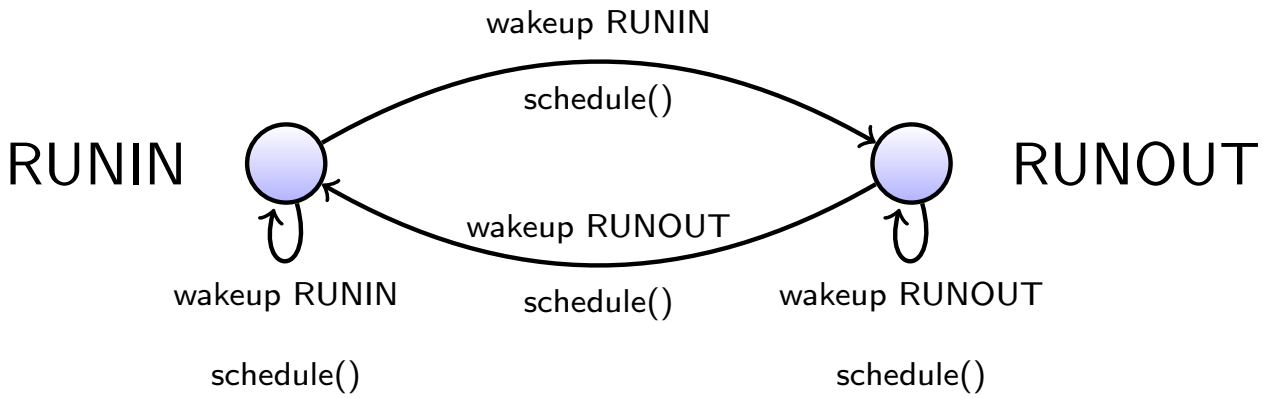
study of the code

- original code is not very readable
- automated formatted with bcpp
- refactored by hand to improve code metrics
  - # of goto: from 11 to 0
  - split sched (100 lines) into 9 functions (and moved to sched.c)
  - used strong typing
- generated documentation with Doxygen

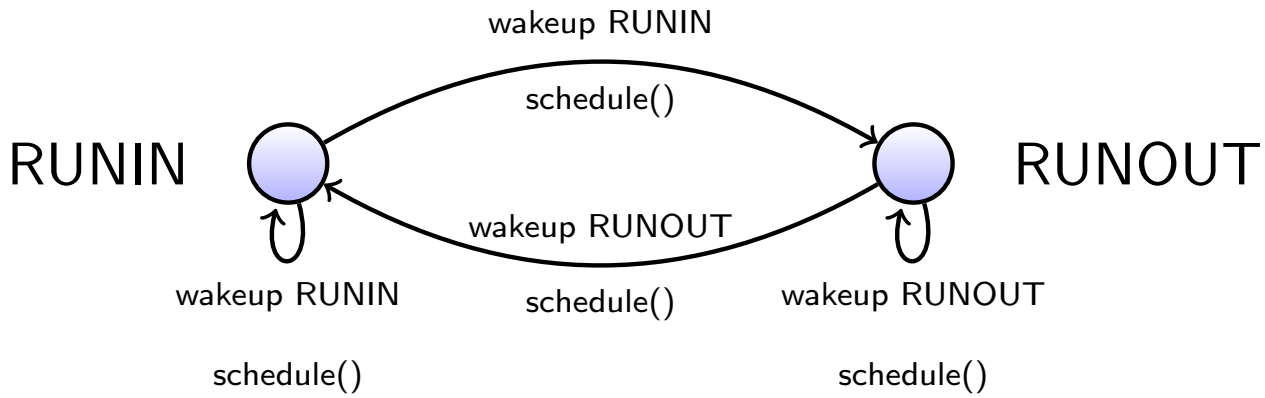
call graph



sched - waiting for changes

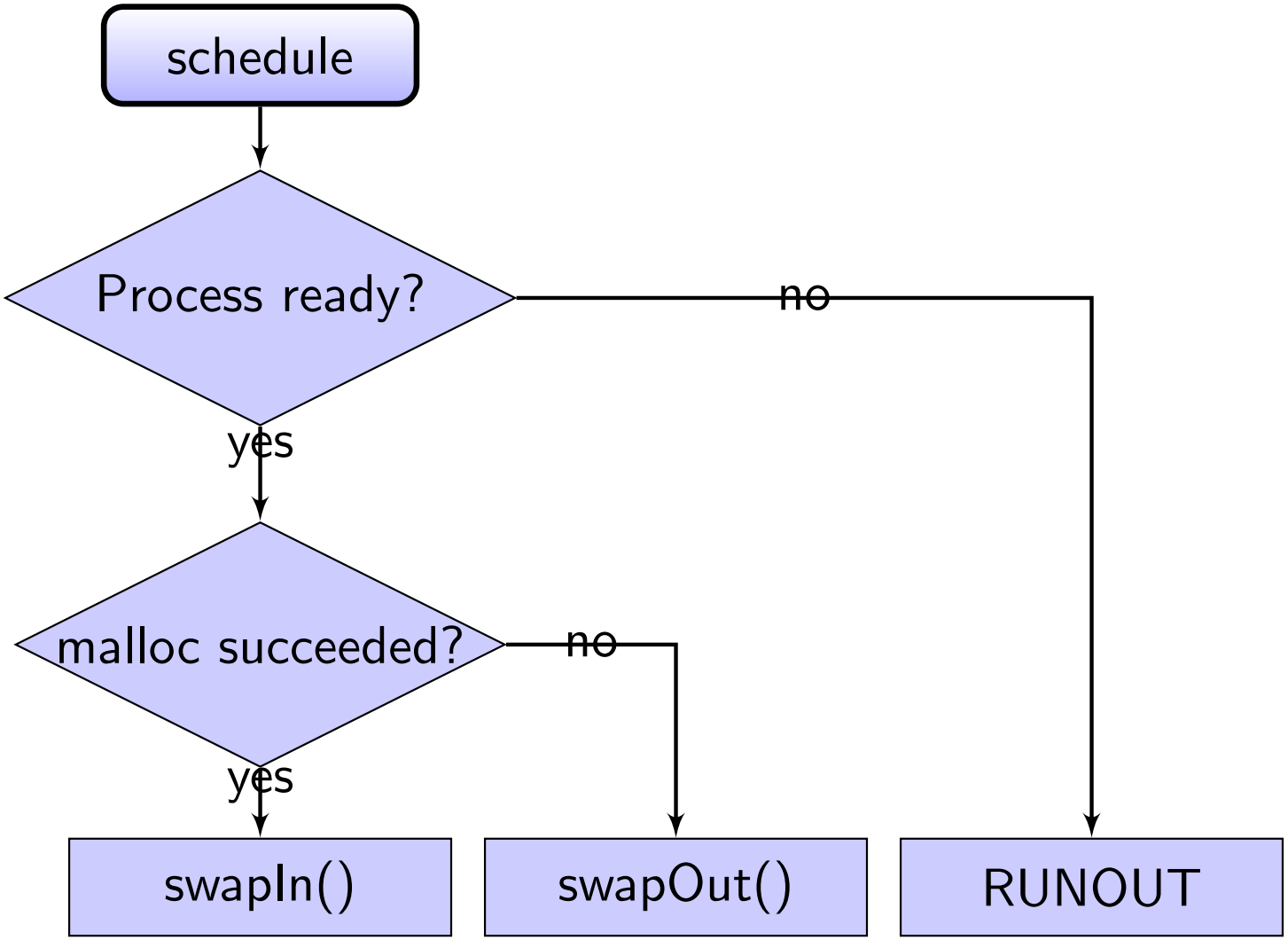


sched - waiting for changes

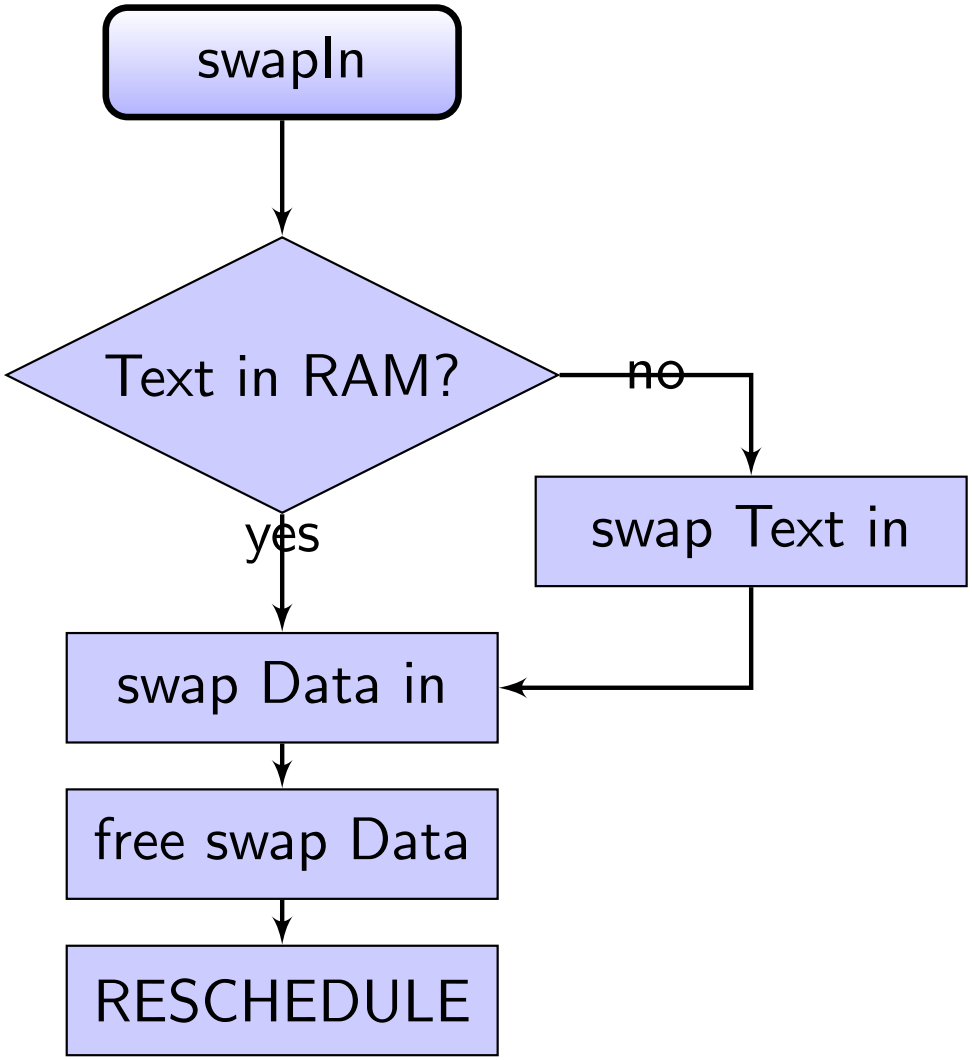


- schedule is looped until it returns something else than RESCHEDULE
  - RUNOUT: no swapped process is ready to run
  - RUNIN: process can't yet be swapped in

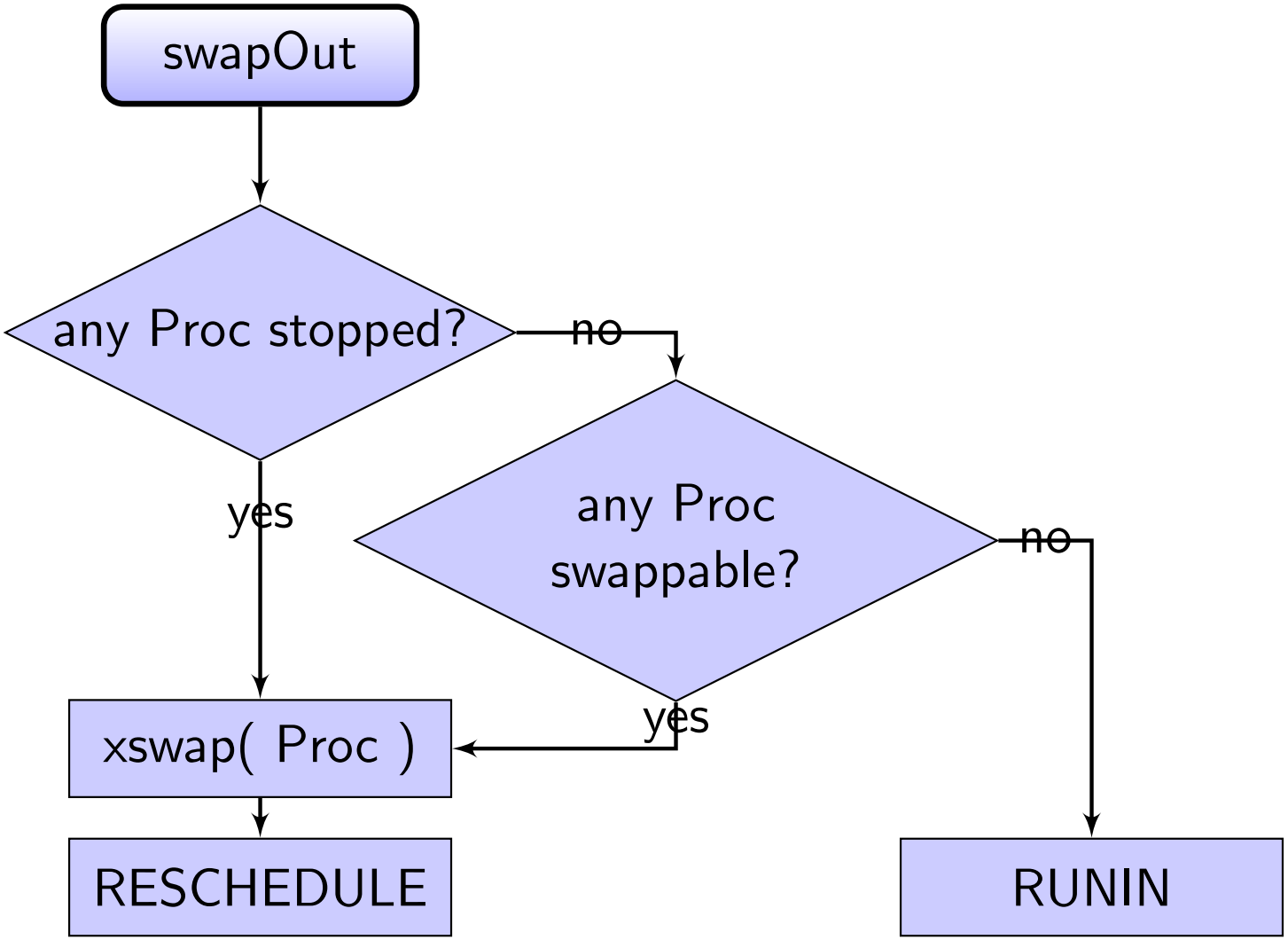
sched : schedule - process swapper



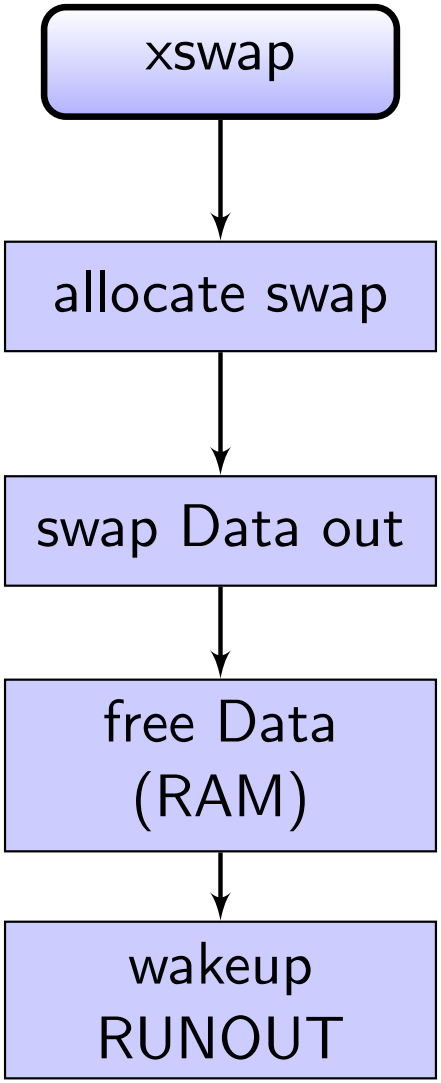
sched : schedule : swapIn



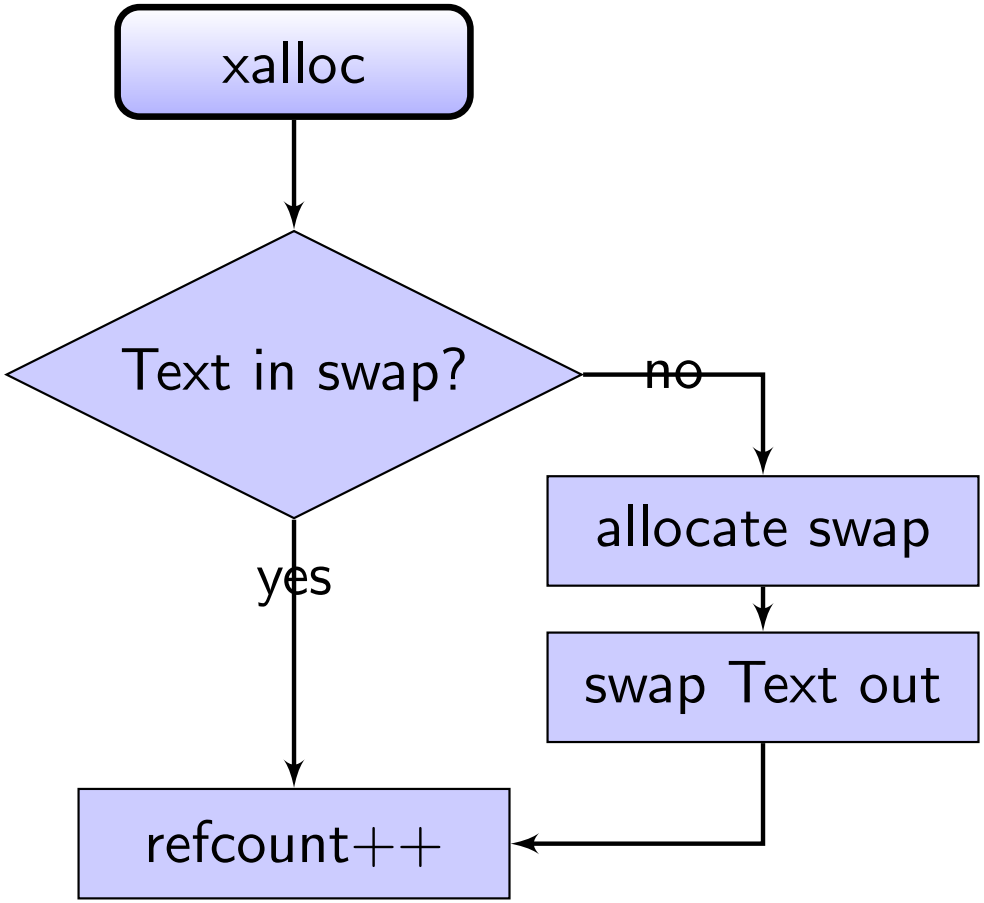
sched : schedule : swapOut



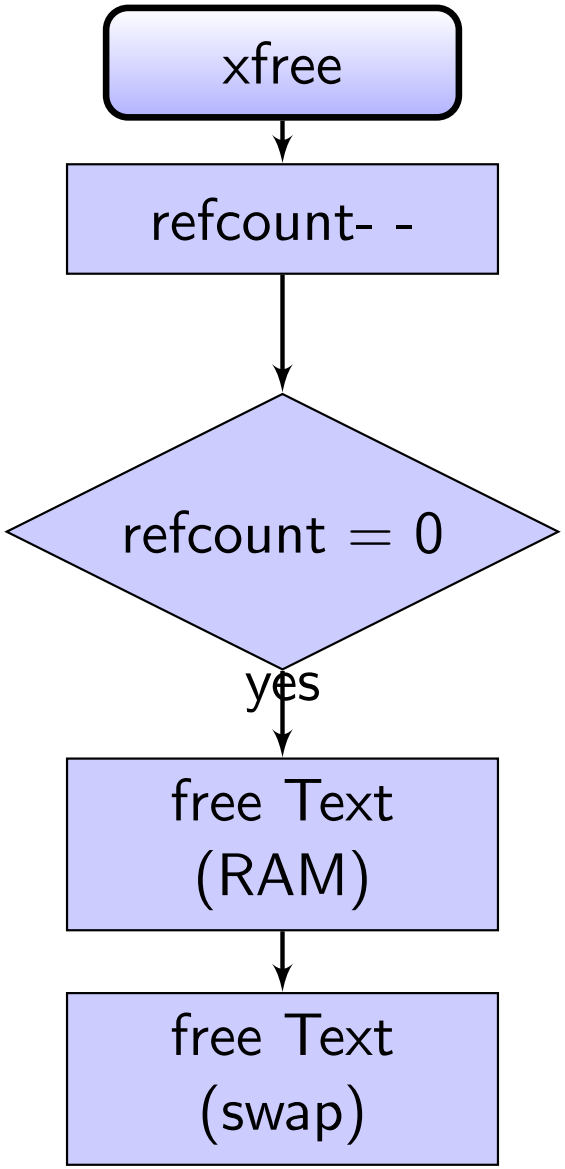
sched : schedule : swapOut : xswap - swap a process out



xalloc - copy text from disc to swap



xfree - remove link to text



comparison to modern coding style

comparison to modern coding style

- don't use goto (at most for exceptions)
- write small subroutines for specific functionality
- use meaningful variable names

comparison to modern coding style

- don't use goto (at most for exceptions)
- write small subroutines for specific functionality
- use meaningful variable names
- use an ANSI C compiler
  - don't optimize (let the compiler do it)
  - use strong typing
  - define a interface (headerfile) to the implementation

about

- find the slides, handout, refactored source and more on `http://n.ethz.ch/~ursf/`