hrwOS

A basic x86 Operating System

Stefan Huber Christian Rathgeb Stefan Walkner

Universität Salzburg – VP Spezielle Kapitel aus Betriebssysteme

January 30, 2007

Overview

Project Management

At the beginning everybody got the exercise to,

1. get an overview of a concrete OS-framework
2. present the other team members the pros and cons.

After we decided to use GeekOS we split up the project into milestones and created an estimated schedule.

Project Management

1. **2006-10-30**: Getting Started
2. **2006-11-08**: ELF Binary Loader
3. **2006-11-22**: Process Management
4. **2006-11-29**: Scheduling
5. **2006-12-20**: Virtual Memory
6. **2006-12-28**: Filesystem
7. **2007-01-03**: IPC
8. **2007-01-08**: Run concurrent application

Basic Facts

hrwOS is an operating system with the following properties:

- Runs on IA-32 hardware (bochs [3]).
- Based on GeekOS 0.3.0 [1].
  $\longrightarrow$ feature complete!
  $\longrightarrow$ And fixed many bugs :-(
- Is therefore developed in C
- User programs written in C / gcc
- About 17.500 lines in *.c files
- About 3.000 lines in *.h files
- Intensive usage of SVN, tagged every project. . .

Feature Outline

- Memory Management
  - **Segmentation + Paging**
  - Swapping
  - LRU like page replacement
  - Unlimited stack size

Feature Outline

- Memory Management
  - **Segmentation + Paging**
  - Swapping
  - LRU like page replacement
  - Unlimited stack size
- Process Management
  - Preemptive scheduling
  - **(Modfied) MLF** and RR schedulers
  - Semaphores (and mutexes in kernel mode)
  - IPC via pipes.

Feature Outline

- Memory Management
  - **Segmentation + Paging**
  - Swapping
  - LRU like page replacement
  - Unlimited stack size
- Process Management
  - Preemptive scheduling
  - **(Modfied) MLF** and RR schedulers
  - Semaphores (and mutexes in kernel mode)
  - IPC via pipes.
- Filesystem
  - GOSFS as special FS in VFS arch.
  - Inode based, up to 4MB files
  - Concurrent access to files via caches
  - Pipes and keyboard/terminal treated as files
    → `echo Hallo | wc`
  - Size of disc is unlimited

Memory Model

Memory model in hrwOS is the suggested one of GeekOS. We use a combination of segmentation and paging.

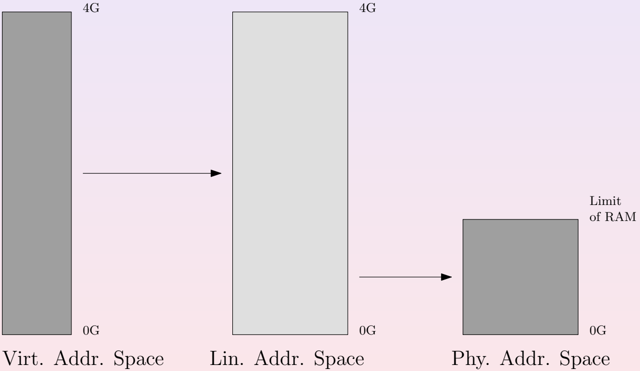

Figure: Intels two step address translation.

Memory Model

The lower 2G is kernel space, the higher 2G user space. The physical memory is mapped one to one into the linear address space which belongs to the kernel.

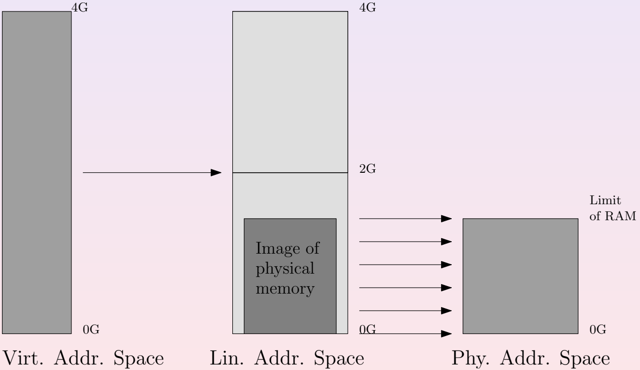

Figure: The physical memory is mapped into lin. addr. space

Memory Model

User space is mapped in the ordinary way to free page frames. Code and data are at lower addresses, stack grows down from top addresses.

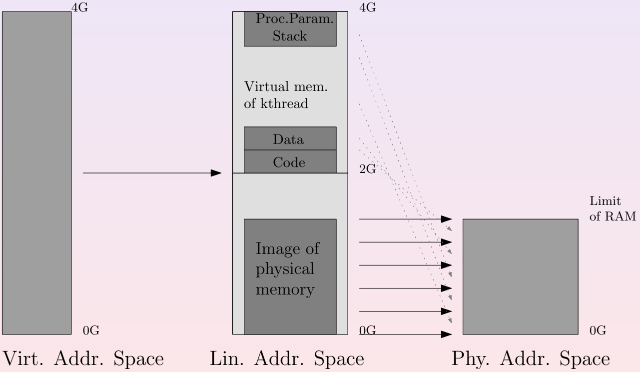

Figure: User space layout.

Memory Model

User segments are mapped to the higher 2G, kernel segments to the whole lin. addr. space.

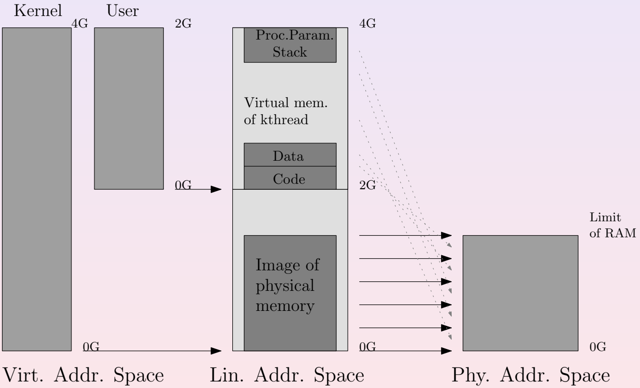

Figure: Segments of kernel and user space.

MLF Scheduler

We use a Multi-Level-Feedback-Scheduler. Currently, there are four ready queues. The higher the index the lower the priority.

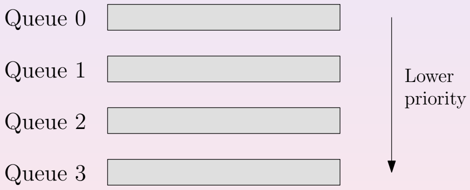

Figure: MLF with four queues

MLF Scheduler

CPU intensive threads are moved to higher indices. Waiting threads to lower ones.

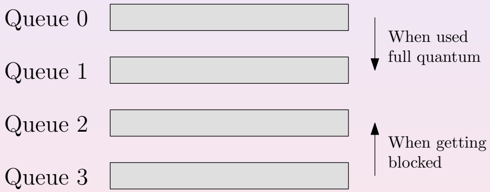

Figure: Transitions in queues

MLF Scheduler

When determining a new thread to run we get the head of a specific queue. To determine the queue we want a perfect mixing sequence of queues:

$$0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0, \ldots$$

MLF Scheduler

When determining a new thread to run we get the head of a specific queue. To determine the queue we want a perfect mixing sequence of queues:

$$0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0, \ldots$$

Queue 0 occurs 50%, Queue 1 occurs 25%, Queue 2 occurs 12.5%, and so on. Furthermore, the mean time of re-occurrence of Queue $n+1$ is double time of Queue n.

MLF Scheduler

This is reached by determining the LSB of an incremented counter which is one. So a schedule is done in $O(1)$ time.

				cnt	queue
0	0	0	**1**	1	0
0	0	**1**	0	2	1
0	0	1	**1**	3	0
0	**1**	0	0	4	2
0	1	0	**1**	5	0
0	1	**1**	0	6	1
0	1	1	**1**	7	0
1	0	0	0	8	3
1	0	0	**1**	9	0
1	0	**1**	0	10	1
1	0	1	**1**	11	0
1	**1**	0	0	12	2
1	1	0	**1**	13	0
1	1	**1**	0	14	1
1	1	1	**1**	15	0

Table: Determine the queue to get new thread from.

Live Presentation...

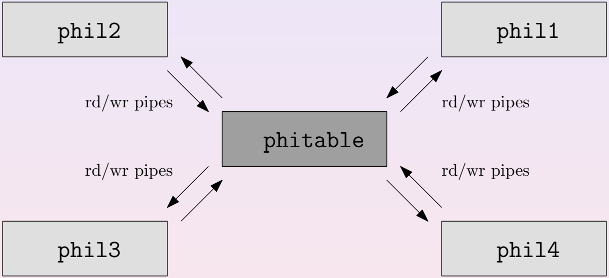

Figure: Server/Client architecture of the Dining Philosophers.

Bibliography I

daveho@users.sourceforge.net.
Geekos website.
http://geekos.sourceforge.net/, 2006.

Stefan Huber, Christian Rathgeb, Stefan Walkner.
hrwos website.
http://www.cs.uni-salzburg.at/~ck/wiki/index.php?n=OS-Winter-2006.HrwOS, 2006.

tbutler@uninetsolutions.com.
Bochs website.
http://bochs.sourceforge.net/, 2006.

Thanks...

...for your attention.