CPSC 457
OPERATING SYSTEMS

FINAL EXAM SOLUTION

Department of Computer Science
University of Calgary
Professor: Carey Williamson

December 10, 2008

This is a CLOSED BOOK exam. Textbooks, notes, laptops, calculators, personal digital assistants, cell phones, and Internet access are NOT allowed.

It is a 120-minute exam, with a total of 100 marks. There are 18 questions, and 11 pages (including this cover page). Please read each question carefully, and write your answers legibly in the space provided. You may do the questions in any order you wish, but please USE YOUR TIME WISELY.

When you are finished, please hand in your exam paper and sign out. Good luck!

Student Name: _____

Student ID: _____

Score: _____ / 100 = _____ %

Multiple Choice

Choose the best answer for each of the following 12 questions, for a total of 12 marks.

1 1. Three **file descriptors** associated with every Linux process are:

(a) standard input, standard output, and standard pipe

(b) **standard input, standard output, and standard error**

(c) standard input, standard output, and standard deviation

(d) standard input, standard output, and standard terminal

(e) standard input, standard output, and standard transmission

1 2. User Mode Linux (UML) is an example of a **virtual machine** environment in which:

(a) Linux runs on top of Windows

(b) **Linux runs on top of Linux**

(c) Windows runs on top of Linux

(d) Windows runs on top of Windows

(e) none of the above

1 3. During the **boot process**, a computer obtains its initial bootstrapping information from:

(a) **a special "boot block" on disk**

(b) the superblock in the root file system

(c) a pre-configured file `vmunix` within the file system

(d) the `/tmp` file system

(e) none of the above

1 4. The **copy-on-write** mechanism provides:

(a) an efficient way to create new processes

(b) a clever way to share virtual memory pages (at least temporarily)

(c) a way to avoid unnecessary page copying

(d) **all of the above**

(e) none of the above

1 5. In memory management, **global** page replacement is usually preferable to **local** page replacement because:

(a) most processes are well-behaved

(b) most processes have small working sets

(c) most processes have large working sets

(d) most processes are highly synchronized

(e) **the set of pages from which to choose is larger**

1 6. Implementing **LRU** precisely in an OS is expensive, so practical implementations often use an approximation called:

(a) MRU

(b) MFU

(c) LFU

(d) LFU with aging

(e) **none of the above**

1 7. For two processes accessing a shared variable, **Peterson's algorithm** provides:

(a) mutual exclusion

(b) progress

(c) bounded waiting

(d) **all of the above**

(e) none of the above

1 8. **Counting semaphores**:

(a) generalize the notion of a binary semaphore

(b) are used for managing multiple instances of a resource

(c) have increment and decrement operations

(d) can use queueing to manage waiting processes

(e) **all of the above**

1 9. The **Banker's Algorithm** is an example of a technique for:

(a) deadlock prevention

(b) **deadlock avoidance**

(c) deadlock detection

(d) deadlock recovery

(e) stabilizing turbulent financial markets

1 10. With **asynchronous I/O**, file system changes will be committed to disk when:

(a) the in-memory inode is updated

(b) **the `sync` daemon runs**

(c) the system administrator feels like doing it

(d) nightly file system backups are run

(e) the system is rebooted

1 11. The operation of **defragmenting** a hard disk:

(a) uses compaction to combat internal fragmentation

(b) **uses compaction to combat external fragmentation**

(c) uses compression to combat internal fragmentation

(d) uses compression to combat external fragmentation

(e) all of the above

1 12. Which of the following is an **idempotent** request?

(a) read the next byte from file `foople`

(b) **read block 3 from file `foople`**

(c) write this block to the end of file `foople`

(d) append file `foople` to file `boople`

(e) link file `foople` to file `boople`

OS Concepts and Definitions

15 13. For each of the following pairs of terms, **identify** the context(s) in which they occur. Then **define** each term and **clarify** the key difference(s) between the two terms.

(a) (3 marks) "host OS" and "guest OS"

```
context:  virtual machines
host OS: underlying OS layer, with access to physical hardware
guest OS: runs on top of host OS, provides services to user,
using the resources provided by the host OS
Virtual machines provide flexible execution environments for users.
```

(b) (3 marks) "page" and "frame"

```
context:  (virtual) memory management
frame:  fixed-size basic unit of physical memory allocation
page:  fixed-size logical unit of process address space;
a page fits in a frame; any page can be put in any frame
```

(c) (3 marks) "reference bit" and "dirty bit"

```
context:  paging-based memory management
reference bit:  indicates if a page has been accessed (recently)
dirty bit:  indicates if a page has been modified (relative to disk)
These bits influence decision-making in page replacement policies
```

(d) (3 marks) "file" and "directory"

```
context:  file systems
file:  a named logical collection of related info, as defined by user
directory:  a logical collection of related files, as defined by user
The directory has metadata about files (name, size, location, etc)
```

(e) (3 marks) "disk partition" and "file system volume"

```
context:  file and storage systems
disk partition:  a logical piece of a disk (or set of disks)
file system volume:  a partition that contains a file system
(as opposed to being empty or used as raw disk or swap space)
A partition can hold a file system
```

Processes

16 14. Answer the following questions about processes.

(a) (4 marks) What is a **process**? What is a **thread**? How are they similar/different?

process: a program in execution
thread: a flow of control within a process
similar: active entities, with many attributes, that consume system resources
different: process is heavyweight, thread is lightweight (part of a process)

(b) (6 marks) There are many **system processes** active on any Linux system. These are typically created at system startup, and operate in the background as daemon processes. Give **three examples** of system (daemon) processes in a Linux system, and briefly state their role in the operation of the system.

Many possible answers here:
init: system initialization, spawns other system processes on boot
swap/pageout: do system paging or swapping when needed
sched: system scheduler
syslogd: log system-related events
sync: periodically flush file system modifications to disk
cron: system timekeeper process (clock, time of day, scheduled jobs)
logind: handle user login events, verify password, launch shell
sshd/telnetd: handle remote terminal sessions
nfsd: handle remote file system requests from clients
ftpd: handle remote file transfer requests from clients

(c) (6 marks) When multiple processes need to cooperate, there is a choice between **shared memory** and **inter-process communication** (IPC). Compare and contrast these two techniques. Make sure to clarify the role of the operating system in each.

shared memory: OS allocates a region of memory that is shared by more than one process (must be on the same machine to do this!). Usually done with page tables; processes can then read/write the shared locations at memory speeds without OS intervention
IPC: message passing; processes communicate using send and receive these are system calls that invoke OS services; the OS is involved in every interaction to copy messages to/from address spaces.
IPC generalizes to processes on different machines using sockets

Memory Management

15 15. Answer the following questions about OS memory management.

(a) (4 marks) One of the design decisions in OS memory management is the choice between **swapping** and **paging**. Define each of these terms, and clarify their respective roles in OS memory management.

swapping: copies entire process image between memory and disk. Assumes contiguous allocation, entire process needed for execution. Used to limit multiprogramming level and avoid thrashing.
paging: divides logical address space of process into fixed-size pieces; process can execute with only a subset of these pages being resident in memory at a time. Provides flexible and efficient memory management, with lots of processes active at a time.

(b) (5 marks) Another key design decision in OS memory management is the choice between **paging** and **segmentation**. Compare and contrast these two approaches to memory management, making sure to identify the strengths and weaknesses of each.

paging: separates physical organization from logical address space
Uses fixed-size units called pages, which are stored in page frames.
Requires page table to keep track of (possibly many) pages.
segmentation: preserves user's structural view of logical address space
Uses variable-size units called segments. Can go anywhere in memory.
Requires segment table to keep track of the (few) segments.
Need base and limit register for each segment.
(These two techniques can also be combined!)

(c) (6 marks) In pure on-demand paging, a **page replacement policy** is used to manage system resources. Suppose that a newly-created process has 3 page frames allocated to it, and then generates the page references indicated below.

(i) How many page faults would occur with **FIFO** page replacement? 12

A B C B A **D A B C D A B** A C B **D**

The **bold font** indicates the references that cause a page fault.

(ii) How many page faults would occur with **LRU** page replacement? 10

A B C B A **D** A B **C D A B** A **C** B **D**

The **bold font** indicates the references that cause a page fault.

(iii) How many page faults would occur with **OPT** page replacement? 7

A B C B A **D** A B **C** D A **B** A C B **D**

The **bold font** indicates the references that cause a page fault.

File and Storage Systems

15 16. Answer the following questions about file systems in general.

(a) (3 marks) In Unix, Linux, and Windows file systems, there are multiple **timestamps** (usually 3) associated with each file. What do each of these timestamps represent?

```
creation:  time when file was first created
modification:  time when file was most recently modified (e.g., written)
access:  time when file was most recently accessed (e.g., read)
```

(b) (6 marks) In class, we discussed three different techniques for organizing the data blocks for each file in a file system, namely **contiguous** allocation, **linked** allocation, and **indexed** allocation. Briefly describe each approach, identifying the strengths and weaknesses of each.

```
contiguous:  allocate file blocks consecutively on disk.
Easy to do sequential or direct (random) access to file.
Simple to implement, but prone to (external) fragmentation,
and very difficult to ‘‘grow’’ file dynamically.
linked:  linked list of available blocks from anywhere on disk.
Each block points to the next.  Ameliorates fragmentation problem,
and easy to grow a file, but random access is difficult and slow
because of many seek times required to traverse chain of pointers.
indexed:  best of both worlds; table of (direct and/or indirect)
pointers to data blocks, which can be anywhere on disk.
Solves fragmentation problem.  Supports sequential and random access.
Can optimize data block layout using cylinder groups, as in Unix.
```

(c) (6 marks) In a storage system with conventional magnetic-media disks, several different **delays** occur when servicing a request. Identify **at least three** of these delays, and comment on their relative contribution to the total delay for servicing a request.

```
seek time:  time to move read/write head from its current position
to the desired track.  May take 5-10 milliseconds (often dominant).
rotational latency:  time for desired sector on target track to spin
under the read/write head.  May take 1-2 milliseconds (medium)
transfer time:  time to read the target sector and transfer bytes
to host computer.  A millisecond or less (low).
wait time:  time spent in I/O queue waiting for service.
Could be 0 of more milliseconds, depending on number of requests
queued and the disk request scheduling policy used.
```

File System Details

12 17. The following page shows some output from some file-system related commands on a local Linux system. Use this output and your knowledge of Linux file systems to answer the following questions.

(a) (1 mark) **How many** different file systems are accessible on this Linux system?
`11`

(b) (1 mark) Which file system is the **fullest** (in terms of percent occupancy)?
`/home/research`

(c) (1 mark) Which file system has the **largest** physical storage capacity?
`/home/scratch`

(d) (1 mark) Which file system has the **fewest** bytes currently stored?
`/boot`

(e) (1 mark) Which disk partition (if any) is being used for **swap space**?
`/dev/sda3`

(f) (1 mark) What is the **type** of the `/tmp` file system?
`ext3`

(g) (1 mark) What is the **type** of the `/proc` file system?
`proc`

(h) (1 mark) How many file systems are remotely mounted using **NFS**?
`7`

(i) (1 mark) Which file system is remotely mounted on server **nsh**?
`/home/grads`

(j) (1 mark) Is this NFS service provided using **UDP or TCP**?
`TCP`

(k) (2 marks) What **block sizes** are used for reading and writing via NFS?
`32 KB for reading, 4 KB for writing`

```
[carey@csl]$ df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda2             20315844   7513824   11753380 39% /
/dev/sda5             10581704   5181092    4854404 52% /tmp
/dev/sda1              1019208     50356     916244  6% /boot
tmpfs                  1684784     77164    1607620  5% /dev/shm
nsi:/export/research 247709760 203879488   31247360 87% /home/research
nse:/export/scratch 3361364788 528637368 2661979928 17% /home/scratch
nsj:/export/proj/dsl 485097928 184513976  275942416 41% /home/dsl
nsg:/export/ug       381885660  79136396  283350608 22% /home/ugc
nsf:/export/ug       381885660  88768580  273718424 25% /home/ugb
nsh:/export/grads    789574392 507946744  241519616 68% /home/grads
nsb:/export/ug       381885660  76958700  285528304 22% /home/uga

[carey@csl]$ cat /etc/fstab
LABEL=/                 /                       ext3    defaults        1 1
LABEL=/tmp              /tmp                    ext3    defaults        1 2
LABEL=/boot             /boot                   ext3    defaults        1 2
tmpfs                   /dev/shm                tmpfs   defaults        0 0
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                   /sys                    sysfs   defaults        0 0
proc                    /proc                   proc    defaults        0 0
LABEL=SWAP-sda3         swap                    swap    defaults        0 0

[carey@csl]$ mount
/dev/sda2 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/sda5 on /tmp type ext3 (rw)
/dev/sda1 on /boot type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
nsi:/export/research on /home/research type nfs (rw,intr,tcp,rsize=32768,wsize=4096)
nse:/export/scratch on /home/scratch type nfs (rw,intr,tcp,rsize=32768,wsize=4096)
nsj:/export/proj/dsl on /home/dsl type nfs (rw,intr,tcp,rsize=32768,wsize=4096)
nsg:/export/ug on /home/ugc type nfs (rw,intr,tcp,rsize=32768,wsize=4096)
nsf:/export/ug on /home/ugb type nfs (rw,intr,tcp,rsize=32768,wsize=4096)
nsh:/export/grads on /home/grads type nfs (rw,intr,tcp,rsize=32768,wsize=4096)
nsb:/export/ug on /home/uga type nfs (rw,intr,tcp,rsize=32768,wsize=4096)
```

General Operating Systems Knowledge

15 18. Throughout CPSC 457 this year, there were several recurring **themes** (i.e., ideas that applied quite broadly across several topics).

(a) (5 marks) One of these themes was **virtualization**. Identify **three** contexts in which virtualization was used as a solution technique. Briefly discuss the technical issues involved, and the benefits of the virtualization approach to the problem.

```
virtual machines:  can run multiple OS on same machine at same time
virtual memory:  separates physical memory management from logical view
virtual file system:  common API for all file systems (local or remote)
Virtualization abstracts away the physical hardware and its constraints,
providing simplicity and flexibility for the users.
```

(b) (5 marks) A second theme was **hardware support**. Identify **three** contexts in which hardware support was used as a solution technique. Briefly discuss the technical issues involved, and the benefits of a hardware-based approach to the problem.

```
multi-processors:  hardware support for parallel computing
multi-core:  hardware support for fine-grain thread concurrency
MMU/TLB: hardware support for address translation, page table lookup
test-and-set, SWAP: atomic hardware instructions for mutual exclusion
protection:  distinguish user-mode priviliges from kernel-mode
Hardware solution provides fast, specialized, high-performance
solution for key OS features and requirements.
```

(c) (5 marks) A third theme was **caching**. Identify **three** contexts in which caching was used as a solution technique. Briefly discuss the technical issues involved, and the benefits of caching as a solution.

```
CPU: on-chip instruction/data caches to avoid memory latency
buffer cache:  in-memory caching to avoid disk latency
disk cache:  remember recent data blocks to avoid disk access latency
write cache:  buffer outgoing writes to improve system performance
storage:  sequential read-ahead and prefetch for I/O controller
file system:  in-memory caching of superblock, inodes, directory info
NFS: client-side caching of info to avoid network latency
Caching optimizes for the common case, and improves performance.
```

*** THE END ***