

CH2

The monolithic structure used in UNIX was easy to implement and maintain.

- ☐ True
- ☒ False

Logging (Accounting) service provided by the OS to allocate resources to each jobs concurrently.

- ☐ True
- ☒ False

Darwin has two different system call types (Mach traps and BSD system calls)

- ☒ True
- ☐ False

Using registers to pass parameters of system call limits the number and length of parameters being passed.

- ☒ True
- ☐ False

Many modern operating systems implements loadable kernel modules

- ☒ True
- ☐ False

System calls are directly accessible by the user program

- ☐ True
- ☒ False

Layered approach simplifies debugging, but layers need to be carefully defined.

- ☒ True
- ☐ False

Program execution is one of the services provided by the OS to the user.

- ☒ True
- ☐ False

FreeBSD is a single-tasking OS.

- ☐ True
- ☒ False

printf() is a system call

- ☐ True
- ☒ False

CH3

What is the output of the following code segment?

```
if (fork() == 0)
```

```
printf ("cc\n");
```

```
else
```

```
{ wait(NULL); printf ("pp\n"); }
```

pp

☐ a. cc

cc

☒ b. pp

☐ c. cc

☐ d. pp

The state transition from waiting to ready happens when a process

☐ a. is dispatched by the scheduler

☒ b. completes an I/O or event handling

☐ c. is interrupted

☐ d. performs an I/O or event handling

The ____ of a process contains temporary data such as function parameters, return addresses, and local variables.

- ☒ a. stack
- ☐ b. data section
- ☐ c. text section
- ☐ d. heap

The list of processes waiting for a particular I/O device is called a(n) ____.

- ☐ a. standby queue
- ☐ b. interrupt queue
- ☒ c. device queue
- ☐ d. ready queue

When a child process is created, which of the following is a possibility?

- ☐ a. The child process runs concurrently with the parent
- ☐ b. the child process has a new program loaded into it
- ☐ c. the child is a duplicate of the parent
- ☒ d. all of the above

The heap (as a part of a process) contains

- ☐ a. temporary data and local variables
- ☐ b. program counter
- ☐ c. the program code
- ☒ d. dynamically allocated memory during run time

A occurs when the CPU switches from one process to another.

- ☐ a. context transfer
- ☒ b. context switch
- ☐ c. CPU switch
- ☐ d. CPU transfer

A process can be terminated due to

- ☐ a. normal exit
- ☐ b. fatal error
- ☐ c. killed by its parent
- ☒ d. all of the above

The state transition from running to waiting happens when a process

- ☐ a. is interrupted
- ☒ b. performs an I/O or event handling
- ☐ c. completes an I/O or event handling
- ☐ d. is dispatched by the scheduler

Which of the following is part of the PCB?

- ☐ a. process ID
- ☐ b. process state
- ☐ c. program counter
- ☒ d. all of the above

CH4

One-to-one thread mapping is less expensive than many-to-many mapping.

- ☐ True
- ☒ False

Threads share the same

- ☐ stack
- ☐ registry
- ☒ code
- ☐ none of the above

Data parallelism

- ☒ distributes subsets of the data across multicores
- ☐ runs thread with data on a single core.
- ☐ distributes threads across cores
- ☐ distributes different operations among multicores.

Thread Local storage for each thread is similar to local variables

- ☐ True
- ☒ False

Deferred cancellation terminates the target thread immediately

- ☐ True
- ☒ False

Modern operating systems extended the process concept to allow a process to have

- ☒ multiple threads of execution and thus to perform more than one task at a time.
- ☐ multiple processes executing simultaneously
- ☐ multiple processes inside a process
- ☐ none of the above

Parallelism means that a system can run more than one task simultaneously

- ☒ True
- ☐ False

Illegal memory access generates synchronous signal.

- ☒ True
- ☐ False

One of the followings is not a challenge for multicore programming

- ☒ scalability
- ☐ dividing activities
- ☐ testing and debugging
- ☐ balance

Multiple threads may not run in parallel on multicore system

- ☒ Many-to-One multithreading model
- ☐ Many-to-Many multithreading model
- ☐ One-to-One multithreading model
- ☐ One-to-Many multithreading model

MEMORY MANGMENT CHAPTER

Consider a logical address space of 64 pages, and page size is 1024 mapped onto a physical memory of 32 frames.

How many bits are there in the physical address?

- ☐ 5 bits
- ☐ 16 bits
- ☐ 10 bits
- ☒ 15 bits

Which page table tracks all physical pages in memory rather than having a page table for each process and keeping track of all possible logical pages?

- ☐ One-level page table
- ☒ Inverted page table
- ☐ two-level page table
- ☐ hashed page table

Consider a logical address space of 64 pages, and page size is 1024 mapped onto a physical memory of 32 frames.

How many bits are there in the logical address?

- ☐ 15 bits
- ☐ 6 bits
- ☐ 10 bits
- ☒ 16 bits

..... allocates the smallest hole that is big enough for a process:

- ☒ Best fit
- ☐ First Fit
- ☐ Worst Fit
- ☐ Last Fit

Swapping is typically supported by mobile systems such as iOS.

- ☐ True
- ☒ False

External fragmentation exists when fixed-size memory blocks (frames) are used.

- ☐ True
- ☒ False

In a system with 32 bit virtual addresses and 4 KB page size, use of one-level page tables for virtual to physical address translation is not practical because of

- ☐ the large amount of external fragmentation
- ☒ the large computation overhead in the translation process
- ☐ the large amount of internal fragmentation
- ☐ the large memory overhead in maintaining page table ✓

Page size is 1024 bytes and process size is 2050. What is the internal fragmentation?

- ☐ 0
- ☐ 1026
- ☒ 1022
- ☐ 2

Logical and physical addresses are the same in compile-time and load-time address-binding schemes.

- ☒ True
- ☐ False

A multilevel page table is preferred in comparison to a single level page table for translating virtual address to physical address because

- ☐ a. It is required by the translation lookaside buffer.
- ☒ b. It reduces the memory access time to read or write a memory location.
- ☐ c. It helps to reduce the size of page table needed to implement the virtual address space of a process. ✓
- ☐ d. It helps to reduce the number of page faults in page replacement algorithms.