

2009.1 Final Exam : LINUX System

Student Id. : (_____) , Name : (_____)

1. (20points) Fill out following blanks.

(1) When a process accesses a virtual address that does not have a valid page table entry, the processor will report a ([1] _____) to Linux.

(2) As there is much less ([2] _____) than ([3] _____) the operating system must be careful that it does not use the ([2] _____) inefficiently.

(3) Each process in the system has its own ([4] _____). These ([4] _____) are completely ([5] _____) from each other and so a process running one application cannot affect another.

(4) if the page has been modified, the operating system must preserve the contents of that pages that it can be accessed at a later time. This type of page is known as a ([6] _____) and when it is removed from memory it is saved in ...

(5) If the *swap algorithm* is not efficient then a condition known as ([7] _____) occurs. In this case, pages are constantly being written to disk and then being read back and the operating system is too ([8] _____) to allow much real work to be performed.

(6) The ([9] _____) contains all the information about the file: the file type, the file's access permission bits, the size of the file, pointers to the data blocks for the file, and so on. Only two items are stored in the directory entry: the ([10] _____) and the ([11] _____).

2. (10points)

(1) Explain in detail how you can correctly implement the command "jobs" in your shell implementation.
(_____)

(2) Explain in detail how you can correctly implement redirection '>' and '<' in your shell implementation.

(i) < :
(_____)

(ii) > :
(_____)

3. (15points) Fill out blanks (1)-(4) in the following source code.

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
#include <errno.h>
#include <sys/wait.h>

int main(int argc, char *argv[]) {
    char *path = "/bin/ls";
    char *arg0 = "ls";
    pid_t pid;
    int pipefd[2];
    int status;

    (1) _____
    pid = fork();
    if (pid == 0) {

        (2) _____
        close(pipefd[0]);
        close(pipefd[1]);
        if (execl(path, arg0, NULL) == -1)
            perror("execl");
    } else {
        if (fork() == 0) {

            (3) _____
            close(pipefd[0]);
            close(pipefd[1]);
            if (execl("/bin/cat", "cat", NULL) == -1)
                perror("execl cat");
        } else {
            close(pipefd[0]);
            close(pipefd[1]);

            (4) _____
        }
    }
}
```

4. (10points) Consider file locking. What is a shared lock (LOCK_SH, also called as read lock)? What is an exclusive lock (LOCK_EX, also called as write lock)? Explain.

(Do not answer like "read lock is used when reading a file, write lock is used when writing a file.")

(1) LOCK_SH : (_____)

(2) LOCK_EX : (_____)

5. (10points)

(1) What information is stored in superblock? List at least five.

(a. _____), (b. _____),

(c. _____), (d. _____),

(e. _____)

(2) Let's say a user wants to read data in the file "/src/code.c". Explain how UNIX OS finds the data location of the file in disk. (Explain the entire process.)

(_____)

6. (15points)

(1)process context consists of (a. _____), (b. _____), (c. _____)

(a. _____) contains (d. _____)

(b. _____) contains (e. _____)

(c. _____) contains (f. _____)

(2) Explain the functionality of fork(). That is, describe what fork() does for creating a new process.

(_____)

7. (10points)

(1a)Between message queue and shared memory, which IPC method is faster? (_____)

(1b) Why? (_____)

(2) What is non-blocking I/O is? Explain.

(_____)

st_mode



8.(10points) Fill out following blanks (1) and (2). (1) should be an octal number
st_mode type variable of a stat structure has information on a file type as shown above.

```
#define S_IFMT (1) _____ // (1) should be an octal number
#define S_IFDIR 0040000 /* directory */
#define S_IFCHR 0020000 /* character special */
#define S_IFBLK 0060000 /* block special */
#define S_IFREG 0100000 /* regular */
#define S_IFLNK 0120000 /* symbolic link */
#define S_IFSOCK 0140000 /* socket */
#define S_IFIFO 0010000 /* fifo */
```

* S_ISDIR macro function checks whether st_mode value has a bit of S_IFDIR constant or not.

Define the macro S_ISDIR by filling out the blanks below. You must use the constant S_IFMT in your answer.

```
#define S_ISDIR(mode) (2) _____
```

* When filling out (1) and (2), you may consider following function that is a part of listfiles.c in our lecture note.

```
/* typeOfFile - return the letter indicating the file type. */
char typeOfFile(mode_t mode)
{
    switch (mode & S_IFMT) {
        case S_IFREG: return('-');
        case S_IFDIR: return('d');
        case S_IFCHR: return('c');
        case S_IFBLK: return('b');
        case S_IFLNK: return('l');
        case S_IFIFO: return('p');
        case S_IFSOCK: return('s');
    }
    return('?');
}
```