

Name: _____

Directions: Neatly write your response to each question in the space provided on the exam or on separate scratch paper. This is an open book/open notes exam worth 100 points. Although it is an open exam I expect you to understand the material and phrase answers in your own words. You will be downgraded if you copy material directly from the book. You have one hour and fifteen minutes to complete the exam. If you finish early, check your work thoroughly and hand in materials including any scratch paper. Then, quietly leave the room. Good luck on the exam.

1. (a) (10 Points) Explain what the v in execvp means.

The command is stored in an array or vector.

- (b) (10 Points) Explain what the p in execvp means.

Search for the command using the PATH environment variable.

2. (10 Points) Write code that causes a jump to point A when SIGUSR1 is raised. Assume that the signal set mask in effect when the code at point A was originally executed returns after the jump is made, and the jump acts like a return from a function with a return value of 3.

```
static sigjmp_buf jumpbuf;
static volatile sig_atomic_t jmpok= 0;
...
void handler() {
    if(jmpok)
        siglongjmp(jumpbuf, 3); f}
...
struct sigaction act;
sig act.sa_handler=handler;
sigemptyset(&act.sa_mask);
act.s_flags=0;
sigaction(SIGUSR1,&act,NULL);
...
val= sigsetjmp(jumpbuf,1);
jmpok=1;
...
```

3. (20 Points) Write code that blocks SIGINT while saving the previous signal set, then unblocks all signals and finally returns to the original signal set plus SIGINT.

```
sigset_t oldmask;
sigset_t newmask;
sigemptyset(&newmask);
sigaddset(&newmask,SIGINT);
sigprocmask(SIG_SETMASK, &newmask, &oldmask);
...
sigprocmask(SIG_UNBLOCK, &newmask, NULL);
...
sigaddset(&oldmask, SIGINT);
sigprocmask(SIG_SETMASK, &oldmask, NULL);
```

4. (20 Points) Write code where a parent process creates a child process. Have the child process put the parent in the background by changing the parent's process group ID to the child's process ID. Then have the parent return itself to the foreground.

```
pid_t child;
...
if (child=fork()) {
    wait(NULL);
    setpgid(getpid(), getpid()); }
else { /* child process */
    setpgid(getppid(), getpid());
    exit(0); }
```

5. (20 Points) Write code where a parent process tests to see whether its only child process is stopped (i.e., the child has raised SIGSTOP). If so, the parent prints out the message, "The child is stopped."

```
int status;
wait(&status);
if(WIFSTOPPED(status)) printf("The child is stopped");
```

6. (10 Points) Write code where a parent process creates a child process and the child process sends SIGUSR1 to the parent process.

```
pid_t child;
if (child=fork())
    wait(NULL);
else { /* child process */
    kill(getppid(),SIGUSR1);
    exit(0); }
```