|  |
| --- |
| 리눅스프로그래밍 01강좌 기말 시험  2022.6.20. 이름 : 학번 : 점수 : |

**※ 다음 리눅스 명령어에 대한 질문에 답하시오.**

**1.** 사용자 user1이 실행중인 프로세스의 수를 확인할 수 있는 명령어를 쓰시오. 2개 이상의 명령어를 | 를 이용해 연결해 보시오.

**2.** 사용자 user1 이 실행중인 프로세스가 다음과 같고 모두 백그라운드(back ground) 프로세스이다. 이중 vi test1 을 포그라운드(fore ground)로 전환하기 위한 명령어를 쓰시오. 여러 명령어를 써야 한다면 순서대로 쓰시오.

|  |
| --- |
| 1653 ttys000 0:00.02 vi test1  1659 ttys000 0:00.00 cat |

**※ 다음 소스 코드는 하나의 부모 프로세스와 두 개의 자식프로세스가 파이프를 통해 통신하는 예제이다. 이를 보고 다음 질문에 답하시오.**

|  |
| --- |
| /\* 헤더 파일 include는 생략 \*/  #define MSGSIZE 100  Int main(void)  {  pid\_t pid1, pid2;  int filedes, p1[2], p2[2];  fd\_set is, ns;  int nread;  char msg[MSGSIZE];  pipe(p2); pipe(p1);  pid1 = pid2 = 0;  pid1 = fork();  if(pid1 > 0)  pid2 = fork();  if(pid1 > 0 && pid2 > 0) {  close(p1[1]); close(p2[1]);  FD\_ZERO(&is);  FD\_SET(p1[0], &is); FD\_SET(p2[0], &is);  ns = is;  while(select(p2[0]+1, &ns, NULL, NULL, NULL) > 0) {  if(FD\_ISSET(p1[0], &ns))  if(read(p1[0], msg, MSGSIZE) > 0)  printf("[parent] %s\n", msg);  if(FD\_ISSET(p2[0], &ns))  if(read(p2[0], msg, MSGSIZE) > 0)  printf("[parent] %s\n", msg);  ns = is;  }  }  else if (pid1 == 0 && pid2 == 0) {  close(p1[0]); close(p2[0]); close(p2[1]);  write(p1[1], "child1", MSGSIZE);  }  else if (pid1 > 0 && pid2 == 0) {  close(p1[0]); close(p1[1]); close(p2[0]);  write(p2[1], "child2", MSGSIZE);  }  else  exit(1);  } |

3. 위의 소스 코드를 pid1에 해당하는 자식 프로세스가 “ls” 명령어를 exec 했을때 출력되는 ls 명령어의 실행 결과를 pid2에 해당하는 자식 프로세스가 읽어서 화면에 출력하도록 수정하시오. 이어서 pid2에 해당하는 자식프로세스는 “complete” 라는 메시지를 부모 프로세스에 전달하고 부모 프로세스는 이를 화면에 출력하도록 수정하시오. (30)

**※ 다음 소스 코드는 socket을 이용한 프로세스간 통신을 하기 위한 코드이다. 다음 질문에 답하시오. (5~6)**

- TCP Socket 서버

|  |
| --- |
| /\* Server, 헤더 파일 include 생략 \*/  int main()  {  int s\_sockfd, c\_sockfd  int server\_len, client\_len  struct sockaddr\_in s\_addr, c\_addr  s\_sockfd = socket(AF\_INET, SOCK\_STREAM, 0);  s\_addr.sin\_family = AF\_INET;  s\_addr.sin\_addr.s\_addr = htonl(INADDR\_ANY);  s\_addr.sin\_port = htons(9734);  server\_len = sizeof(s\_addr);  bind(server\_sockfd, &s\_addr, server\_len);  listen(s\_sockfd, 5);  while(1) {  char msg[256];  client\_len = sizeof(client\_address);  c\_sockfd = accept(s\_sockfd,  (struct sockaddr \*)&c\_addr, &client\_len);  read(c\_sockfd, msg, 256);  write(c\_sockfd, msg, 256);  close(c\_sockfd);  }  } |

- Socket 클라이언트

|  |
| --- |
| /\* TCP Client, 헤더파일 include 생략 \*/  int main()  {  int sockfd  int len  struct sockaddr\_in address;  int result;  char msg =“Hello”';  sockfd = socket(AF\_INET, SOCK\_DGRAM, 0);  address.sin\_family = AF\_INET;  address.sin\_addr.s\_addr = inet\_addr("127.0.0.1");  address.sin\_port = htons(9734);  len = sizeof(address);  result= connect(sockfd, (struct sockaddr\*)&address, len);  if(result == -1) exit(1);  write(sockfd, msg, 256);  read(sockfd, msg, 256);  close(sockfd);  exit(0);  } |

4. 이 서버/클라이언트 프로그램은 정상적으로 동작하지않는다. 잘못된 부분을 수정하여 정상 동작하도록 만들어라. 특히, 서버는 클라이언트가 전송한 메시지를 받으면 화면에 출력하도록 하라. (10)

5. 서버 프로세스가 세션이 끊겨도 종료되지 않고 deamon 화 되도록 코드를 수정하시오. (20)

6. 서버 프로그램이 SIGINT 시그널을 받았을 때 “SIGINT is ignored” 라는 메시지를 화면에 출력하도록 코드를 수정하시오. (20)