**UNIX 프로그래밍 중간시험 2021-10-25**

분반: 1 학과: \_\_컴퓨터과학과\_\_\_ 학번: \_\_201810993\_\_\_ 이름: \_\_\_\_정경재\_­­­\_\_\_\_

다음 문제를 풀어 답을 쓰거나 캡처를 달고, “2021-UNIX-mid-학생이름.doc”(또는 docx, hwp)로 저장하여 제출하시오.

프로그램 소스파일은 각 문제에 표시된 이름으로 저장하고 제출하시오.

(comment와 출력에 한글, 영문 다 써도 됩니다.)

시험 중 문의는 email: [ybaek@smu.ac.kr](mailto:ybaek@smu.ac.kr), 카카오톡 또는 문자 010 6277 3268로 하세요.

제출 마감은 10월 25일 22시 00분입니다.

0. 아래 문장을 읽으며 부정 방지 의지를 다져 주세요.

“나는 오늘 유닉스 프로그래밍 중간 시험을 온라인으로 응시함에 있어, 일체의 부정행위를 하지 않고, 오직 나의 능력과 양심을 바탕으로 답안을 작성하여 제출할 것임을 엄숙하게 서약합니다. 2021년 10월 25일 상명대학교 ㅇㅇㅇㅇ학과 ㅇㅇㅇ(이름)”

1. vi와 gcc를 이용하여 C 프로그램을 개발하는 과정을 자세히 설명하시오. (10)

(여기에 설명, 그림을 그려도 됩니다.)

C 프로그램을 생성하기 원하는 곳에 vi 파일명.c를 한다면 다음과 같은창이 나옵니다. 텍스트, 모니터, 전자기기, iPod이(가) 표시된 사진

자동 생성된 설명

그리고 우리가 원하는 c 프로그램을 작성합니다.  
예를 들어 hello라는 문자열을 출력하는 프로그램을 아래의 사진과 같이 작성했다면

텍스트, 모니터, 전자기기, 스크린샷이(가) 표시된 사진

자동 생성된 설명  
:WQ를 통해 해당 내역의 파일을 저장하고 gcc complier를 통해 실행 가능한 코드를 만듭니다.  
gcc -o (output의 파일명) (compile할 파일의 이름)

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

이렇게 코드가 만들어졌다면 이를 ./(실행하고자 하는 파일의 이름)을 통해 C 프로그래밍을 할 수 있습니다.

2. stdin, stdout, stderr에 대해 설명하고 이것을 바탕으로 I/O redirection을 설명하시오. (10)

(여기에 설명, 그림을 그려도 됩니다.)

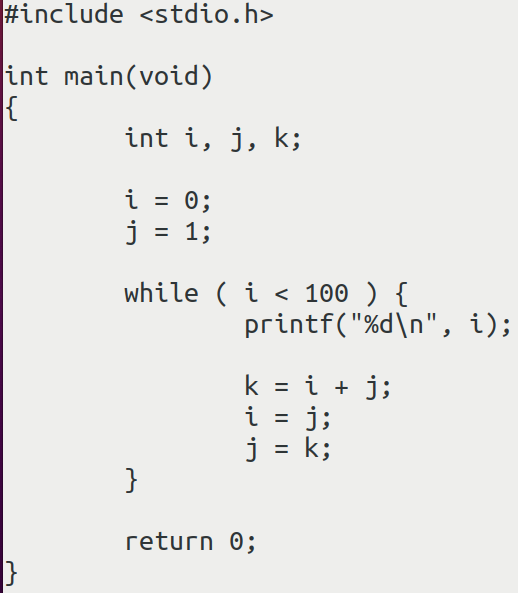
Stdin, stdout, stderr은 기본 파일 기술자 입니다.

파일 기술자는 0번부터 시작되는 데 0번은 stdin(표준입력), 1번은 stdout(표준 출력), 2번은 stderr(표준 오류)를 의미합니다. 그래서 대체로 0번은 키보드의 입력, 1,2번은 모니터화면을 이야기하지만 이는 사용자가 변경할 수 있습니다. 프로세스는 파일을 열을 때는 0부터 시작해 증가하는 순서로 되어있고 이미 기술자가 있는번호는 넘어갑니다. 그래서 기본 기술자만 있는 상태에서 저수준의 open으로 파일을하나를 읽으면 이 open은 4번을 부여받게 되는 방식으로 되어있습니다. 하지만 아까도 말한 것처럼 i/o redireaction을 변경할 수 있습니다. Close를 통해 stdout을 제거하고 이를 open을 통해 파일에 쓰는 작업을 한다면 출력은 open 한 파일에 저장됩니다. 이러한 방식으로 open 과 close를 통해 I/O의 방향을 변경 할 수 있습니다.

3. 다음의 C 프로그램이 어떤 일을 하는 가를 설명하고, 이와 같은 역할을 하는 bash 스크립트를 3.sh로 작성하고, 실행하여 결과를 보이시오. (10)(수행 과정과 수행을 캡처하여 붙이시오. 소스 파일 제출합니다. 소스에는 comment를 넣어야 합니다.)

해당 프로그램은 피보나치 수열을 출력하는 일을 하는 프로그램입니다.

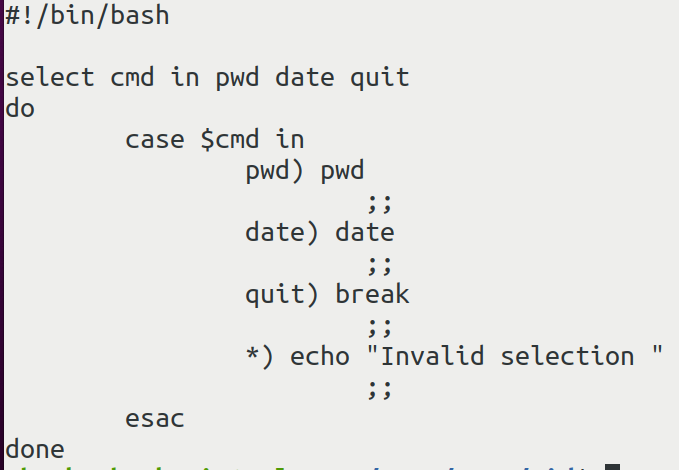
피보나치 수열은 1 1 2 3 5 13 21 34 55 89 ~로 동작하는 n이 3부터라고 했을 경우  
n = (n-2) + (n-1)인 식을 가지는 수열입니다.



텍스트이(가) 표시된 사진

자동 생성된 설명

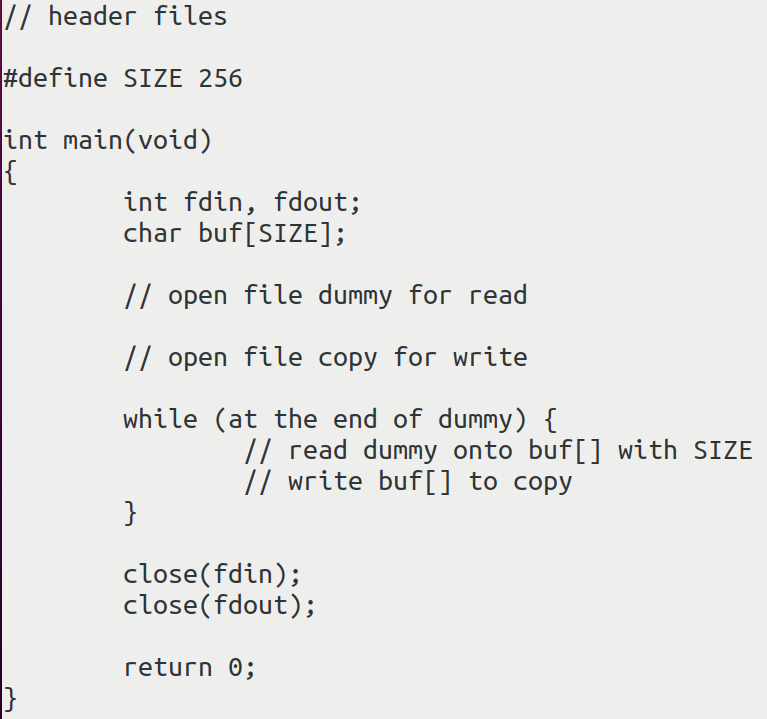
4. 다음의 bash 스크립트와 같은 역할을 하는 C 프로그램을 4.c 로 작성하고, 실행하여 결과를 보이시오. (10)(컴파일과 수행을 캡처하여 붙이시오. 소스 파일 제출합니다. 소스에는 comment를 넣어야 합니다.)



텍스트이(가) 표시된 사진

자동 생성된 설명

5. 다음은 ‘시스템 호출’을 사용하여 dummy라는 파일을 copy로 복사하는 프로그램의 골격이다.



이것은 5.c로 완성하고, 크기가 100M인 파일 dummy를 이용하여 수행 결과를 보이시오. (10) (모든 과정과 수행을 캡처하여 붙이시오. 소스 파일 제출합니다. 소스에는 comment를 넣어야 합니다.)

텍스트, 모니터, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명

6. 5.c를 매크로 정의 SIZE를 변화시키며 이 크기가 수행 시간에 어떤 영향을 주는지 알아보려고 한다. (10)

(1) SIZE가 256으로 컴파일하고 수행한 후

$ time ./5 로 시간을 측정하고 그 결과를 보이시오. (결과 캡처)

텍스트이(가) 표시된 사진

자동 생성된 설명

(2) SIZE가 4096으로 컴파일하고 수행한 후

$ time ./5 로 시간을 측정하고 그 결과를 보이시오. (결과 캡처)

텍스트이(가) 표시된 사진

자동 생성된 설명

(3) 왜 이런 결과가 나오는지를 설명하시오.

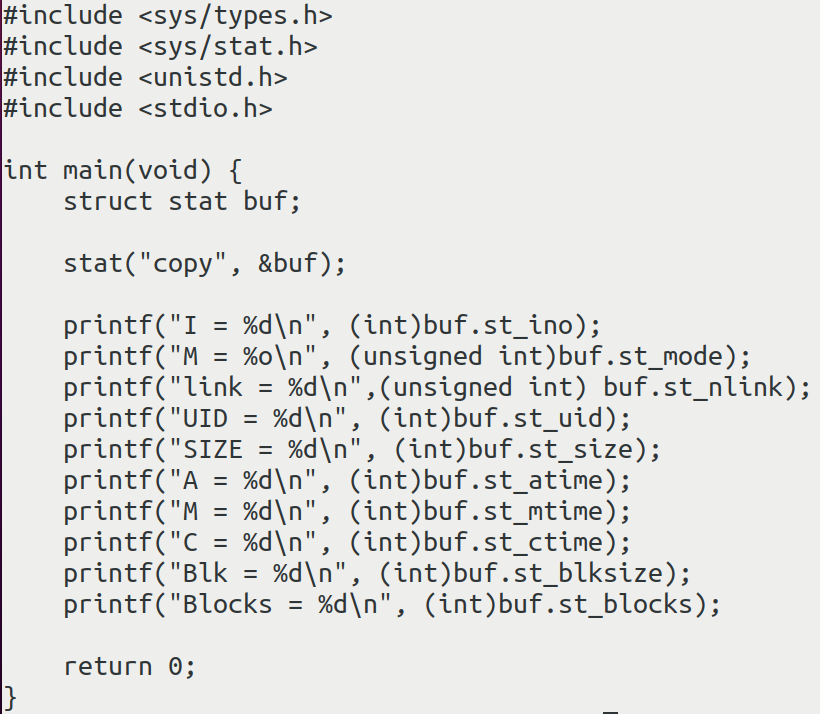
해당 결과가 나오는 이유는 다음과 같습니다.  
첫번째 SIZE가 256일 경우는 256바이트만큼 읽고 쓰는 작업을 진행하고 4096인 경우는 4096바이트만큼 읽고 쓰는 작업을 하므로 이 둘의 차이는 28과 212승 즉 24씩 매 번 읽고 쓰는 차이가 벌어져 시간의 차이가 발생하게 됩니다.

(4) 가장 최적의 SIZE를 찾을 수 있는지, 있다면 얼마인지 쓰고, 그 이유를 설명하시오.

가장 최적의 SIZE는 4096입니다.

왜냐하면은 dummy파일과 copy파일의 정보에서 파일의 내용을입출력할 때 사용하는 버퍼의 크기가 4096으로 되어있으므로 4096을 넘는 값으로 SIZE를 지정해도 4096씩 끊어 넣으므로 4096이 최적의 크기입니다.

7. 다음 프로그램을 7.c로 만들고 수행하여 결과를 보이시오.



결과를 캡처하여 붙이고, 결과의 모든 숫자가 의미하는 바를 자세하게 설명하시오. (10)

텍스트이(가) 표시된 사진

자동 생성된 설명

I = Inode의 값을 말하고

M = 파일의 mode의 값을 말합니다. 10644의 권한을 가진 파일입니다.

Link = 하드링크의 갯수를 말합니다.

UID = 파일의 소유자의 ID를 말합니다.

SIZE = 파일의 크기를 의미합니다.

A = 마지막으로 파일을 읽거나, 실행한 시각입니다.

M = 마지막으로 파일의 내용을 변경한 시각입니다.

C = 마지막으로 inode의 내용을 변경한 시각입니다.

Blk = 파일의 내용을 입출력할 때 사용하는 버퍼의 크기입니다.

Blocks = 파일을 512 바이트씩 블록으로 나눈 갯수를 저장한다.

8. 시스템 시간 관련 시스템 콜 time()과 함수 localtime()만을 이용하여 명령어 date를 수행한 것과 같은 결과를 만드는 프로그램을 8.c로 작성하고, 실행하여 결과를 보이시오.

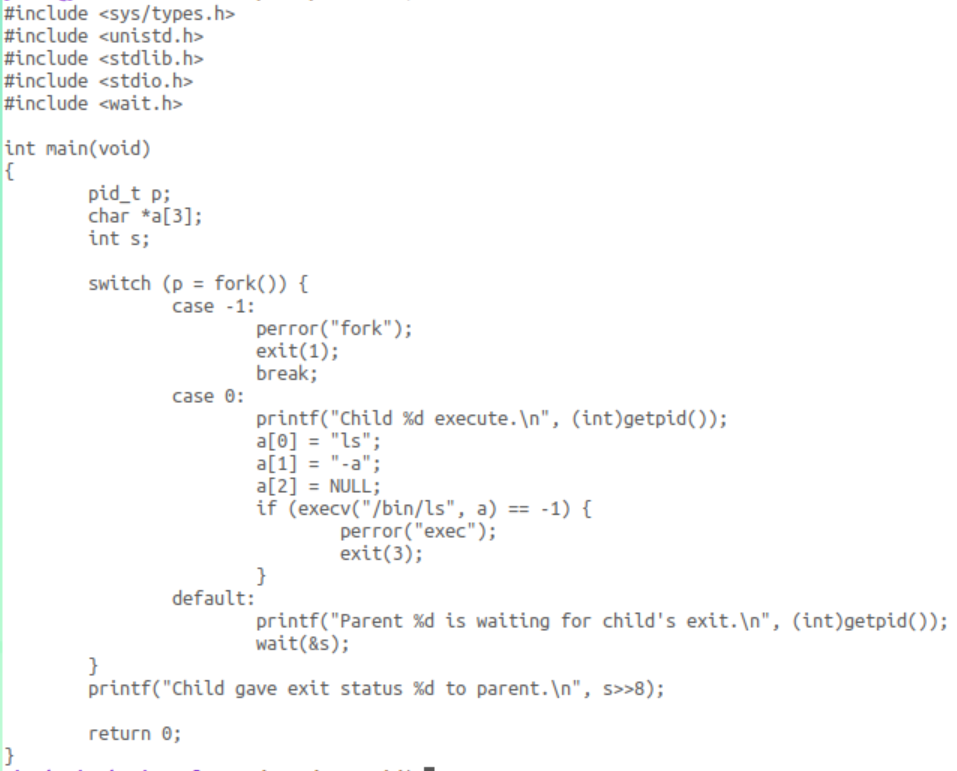


(10)(수행을 모두 캡처하여 붙이시오. 소스 파일 제출합니다. 소스에는 comment를 넣어야 합니다.)

텍스트이(가) 표시된 사진

자동 생성된 설명

9. 다음 프로그램에 사용된 시스템 콜과 함수 fork(), execv(), wait(), exit(), perror()에 대하여 자세히 설명하시오. (10)



이 함수에서 사용된 fork()함수는 새로운 자식 프로세스를 생성하는 함수로 부모 프로세스가 호출하여 자식 프로세스를 생성합니다. 이때 생성된 자식 프로세스는 부모 프로세스의 메모리를 복사해 가져옵니다. Fork()를 호출한 부모는 자식의 프로세스번호를 받고 자식은 0을 받습니다.  
execv()는 자식 프로세스에서 실행이 됩니다. 이를 실행하게 된자식은 /bin/ls의 경로의 ls -a라는 명령어를 실행하는 프로세스로 변경이 됩니다.  
wait()는 자식의 프로세스가 종료하길 기다리는 프로세스 동기화 함수입니다.  
따라서 부모가 wait를 호출하기 전에 리턴을 했다면 wait는 즉시 리턴을 하고 아니면 자식의 프로세스의 PID를 리턴받거나 리턴이 -1이면 자식이 없다는 뜻입니다.  
perror()은 오류 메시지를 출력하는 함수입니다.

10. find와 grep 명령은 소스코드 분석 시에 유용하게 사용되는 것들이다. 이들에 대해 조사하고 사용 예를 보이시오. (10)(설명하고 사용 예 캡처)

Grep 명령어는 파일이나 표준 입력을 검색하여 주어진 정규 표현식과 맞는 줄을 찾아 프로그램의 표준 출력으로 출력하는 명령어입니다.

Find 명령어는 하위 폴더에 존재하는 파일 및 디렉토리를 검색하는 명령어 입니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

(정리) 제출해야 할 파일

답안파일 2021-UNIX-mid-학생이름.doc

소스파일 3.sh, 4.c, 5.c, 8.c

를 e-Campus 중간시험과제 란에 제출하시오.

(끝, 수고했습니다.)