1.10. 연습문제

1. 터미널에서 “man ar”을 타이핑하여 ar 명령에 대한, 설명을 읽어 보도록 한다. 사용할 수 있는 option 들에 관하여 알아본다.

2. double 형의 변수 x에 값을 입력하여, mathematical library (libm.a)에 있는 sqrt(x)를 수행하여 결과를 출력하는 main 프로그램을 작성하여 실행하도록 한다. 이 프로그램을 컴파일 하려면, libm.a를 포함하여 컴파일 하여야 한다. (man을 활용하도록 한다.) (use -l option)

3.
```
double mysquare(double d) {
        return (d x d);
}
```
위의 프로그램을 mysquare.c 파일에 작성하여, libmysquare.a라는 static 라이브러리를 생성하도록 하라.

4. 3번 문제에서 작성한 라이브러리의 mysquare () 함수를 이용하여, “2.0” 값의 square 값을 출력하는 main 프로그램을 작성하고 컴파일하여 실행하도록 한다.

5. 사용자가 입력한 단어를 single linked list에 저장하고 출력하는 프로그램을 작성하라.

addition: 단어를 입력하여, 링크에 저장

print: 현재의 linked list에 있는 모든 단어를 출력

```
% a.out
    Input Command: 1 (addition), 2(print)
    1
    input a word: This
    Input Command: 1 (addition), 2(print)
    2
    This
    Input Command: 1 (addition), 2(print)
    1
    input a word: is
    Input Command: 1 (addition), 2(print)
    2
    This    is
    Input Command: 1 (addition), 2(print)
```

2.9. 연습문제

1. argv[1] file의 내용을 역순서로 argv[2] 파일에 저장하는 invert 프로그램을 작성하라. 예를 들면, file1의 내용이 “Hello, World”인 경우, 사용자가 “% invert file 1 file2”라고 수행하면, file2가 생성되며 (file2가 이미 존재하는 경우에는, file2의 내용이 삭제됨) file2의 파일에는 “Hello, World”의 내용을 역순으로 작성한 “dlroW ,olleH”가 저장된다.

   - 따라서, 사용자가 다음과 같이 순차적으로 프로그램을 실행하면, file1의 내용과 file3의 내용은 동일하게 된다.

     % invert file1 file2

     % invert file2 file3

2. argv[1]의 파일을 내용을 읽어서 홀수번째의 바이트들의 내용을 argv[2]에 저장하고, 짝수번째의 바이트들을 argv[3]에 저장하는 프로그램을 lseek를 이용하여 작성하라. argv[2], argv[3] 이름의 파일이 있는 경우, 이 프로그램은 실패하도록 한다. (예를 들면, argv[1] 파일의 내용이 “0123456789” 인경우, 이 프로그램을 **% a.out file1 file2 file3** 이렇게 실행하면 argv[2] 파일에는 “02468”이 저장되고, argv[3] 파일에는 “13579”가 저장된다.)

3. 2번에서 작성한 프로그램을 실행하여 생성된 두 개의 파일을 다시 원래의 파일로 만드는 프로그램을 작성하라. (예를들면, file2에는 “02468”, file3에는 “13579”의 내용이 기록되어 있는 경우, **% a.out file2 file3 orignal_file** 이렇게 실행하면 original_file의 내용은 file1의 내용과 동일하게 “0123456789”가 된다.

4.7. 연습문제

1. Child process를 생성하여 생성된 child process가 “ls –l”을 실행하는 main 프로그램을 작성하라. 이때, parent process는 생성된 child prcess가 exit 할 때까지 기다린 후 child process의 exit code에 따라서 child process가 정상적으로 수행되고 종료하였는지 또는 비정상적으로 종료하였는지를 출력하도록 한다.

2. Parent process는 argv[1]을 입력으로 받는다. Parent는 child를 생성하여, child가 /bin 디렉토리에 있는 기존의 cat 프로그램과 유사한 동작을 실행하도록 하며, 이때 parent가 child를 실행하는 방법은 다음과 같다.

```
execl (“./mycat”, “myCat”, argv[1], (char *)0);
```

Parent는 child가 정상적으로 종료하였는지의 여부를 출력하도록 한다. Parent와 child 프로그램을 작성하라.

3. 기존의 shell과 유사하게 동작하는 myShell.c 프로그램을 작성하라. 프로그램은 무한 loop를 돌면서 화면에 프롬트 (예들들면 **%**)를 출력하고, 사용자가 명령을 입력하면 child를 생성하여 child가 해당 명령을 수행하도록 한다. 사용자가 입력한 명령은 /bin, /usr/bin, /usr/local/bin 디렉토리 내에 있는 명령으로 제한한다. (프로그램에서는 읽은 명령이 위 3개의 디렉토리 중 어느 디렉토리에 있는가를 확인하여야 한다. 이때 사용하는 함수는 access(2)를 사용하도록 한다. access(2)의 사용방법은 man을 이용하거나, 인터넷에서 찾아보도록 한다.) 만약 사용자가 입력한 명령이 위의 3 디렉토리에 없는 경우 프로그램에서는 “command not found”를 출력한다.

4. 3번에서 작성한 프로그램에 다음과 같은 기능을 추가하라. 3번에서는 사용자가 입력한 명령을 찾는 디렉토리가 기본적으로 3개로 고정되어 있었다. 본 프로그램에서는 사용자가 setpath 명령으로 이 디렉토리를 추가할 수 있도록 한다. 따라서, myShell 프로그램이 처음으로 실행될 때는 기본적으로 찾는 디렉토리가 없으며, 사용자가 **setpath /bin** 이라고 실행하면 /bin 디렉토리가 기본적으로 찾는 디렉토리에 추가되는 방식이다. 이 경우 기본적으로 찾는 디렉토리가 3번 프로그램과 같이 되려면, 사용자가 “**setpath /bin**” “**setpath /usr/bin**” “**setpath /usr/local/bin**” 명령을 차례로 수행하여야 한다.

5. 4번에서 작성한 프로그램에 다음과 같은 기능을 추가하라. 홈 디렉토리의 .myShellRc 파일에는 필요한 setpath 명령이 저장되어 있으며, myShell 프로그램이 실행될 때, 이 파일의 내용을 자동적으로 수행할 수 있도록 한다.

6.5. 연습문제

1. 다음과 같은 기능을 수행하는 pipeserver.c 와 pipeclient.c 프로그램을 작성하라.

   - pipeserver.c : The program reads what is written to a named pipe and writes it to standard output.
   - pipeclient.c : The client writes an argv[1] file contents to a named pipe.

   - pipeserver가 초기화 과정에서 생성한 named pipe 이름은 public 하게 알려져 있다.
   - pipeserver는 종료하지 않는 daemon process 형식으로 동작한다.
   - Client는 argv[1] 파일의 내용을 named pipe에 write하고 종료한다.