

## 2017년도 프로그래밍응용및실습 프로젝트 (영화 관리 프로그램)

- ◎ 목적 : 영화 관리 프로그램 개발을 통해 동적 메모리 할당, 자기 참조 구조체, 파일 입출력 사용 능력 배양
- ◎ 프로젝트는 팀으로 수행해야 하며, 단독 또는 팀은 있으니 혼자 100% 참여로 제출할 경우 획득 점수의 50%만 부여
- ◎ 제출물 :
  - 1. 출력물 : 보고서, 평가표
    - 보고서에는 프로그램 개요, 프로그램 설계서(함수목록 및 함수설명서), 프로그램 소스, 실행 화면, 팀원의 작업 분배내용, GitHub 활동 내용 등이 포함되어야 함
    - 평가표에는 팀원의 참여 비율을 명확히 명시해야 하고, 구현 여부 항목에는 완전히 구현했으면 O, 구현하지 못했거나 부분적으로 구현했으면 X를 표기해야 함. 부분구현과 미구현은 점수 없음. 완전히 구현하지 못했는데 O로 표기하면 그 항목은 0점 처리 함
  - 2. 온라인 제출물(스마트 캠퍼스) : 보고서, 평가표, 프로그램
    - 모든 파일의 이름은 팀장의 학번으로 함
    - 보고서는 아래한글로 작성하여 한글 파일로 제출함
    - 평가표는 엑셀 파일로 제출 함
    - 프로젝트 프로그램은 프로젝트를 위한 디렉터리를 새로 만들고, 그 안에 프로그램 소스코드와 makefile을 저장한 후 그 디렉터리를 zip으로 압축하여 업로드해야 하며, 디렉터리 이름과 압축 파일명은 팀장의 학번으로 함
- ◎ 제출일 : 12월 10일 23:59, 출력물은 12월 11일 실습시간에 제출
- ◎ 카피본 발견 시 소스 제공 팀과 함께 0점 처리, 프로젝트 검사 시 각 참여자가 작업 분배 항목에 관한 질문에 답변을 못할 경우 0점 처리
- ◎ 프로젝트 요구 조건

- 1. 라이브러리 함수를 최대한 사용하고 라이브러리에 없는 기능은 직접 코딩해야 함
- 2. 프로그램은 디폴트로 다음과 같은 6개의 파일을 사용함
  - movie\_log, movie\_list, director\_log, director\_list, actor\_log, actor\_list
- 2.1. movie\_log 파일
  - 영화 목록의 모든 기록을 담는 파일로 각 영화에 대한 정보를 다음과 같은 형식으로 저장함. 각 항목은 :(콜론)으로 구분함
    - tag:serial\_number:title:genre:director:year:time:actors
  - \* tag : 이 레코드의 생성 형태를 알려주는 것으로 add, update, delete가 올 수 있음. 각각은 다음과 같은 의미를 가짐
    - . add : 새로운 영화를 삽입함
    - . update : 기존의 영화 정보를 수정함
    - . delete : 기존의 영화 정보를 삭제함
  - \* serial\_number : 영화의 고유 관리번호. 1부터 부여되며 새로운 영화가 추가(add)될 때마다 1씩 자동으로 증가됨. 삭제된 영화에 할당된 번호는 재사용하지 않음

- \* title : 영화 제목
- \* genre : 영화 장르
- \* director : 감독 이름
- \* year : 제작 년도
- \* time : 총 상영시간(분)
- \* actors : 등장 배우들로 “,”로 분리되어 들어감
- movie\_log 파일의 예는 다음과 같음
  - add:377:Matrix Reloaded:Sci-Fi:Wachowski Brothers:2003:138:K. Reeves,\
  - Fishburne, C.A. Moss, H. Weaving, Jada Pinkett-Smith, Gloria Foster,\
  - M. Bellucci, Daniel Bernhardt
  - update:371:::Jason Statham, Wesley Snipes, Ryan Phillippe,\
  - Justine Waddell, Henry Czerny, Nicholas Lea, Jessica Stehen
  - delete:354:::==
- \* 예제에서 각 행의 \는 다음 줄로 계속 됨을 의미하는 것으로 movie\_log 파일에는 없는 것임
- \* 첫 번째 레코드는 tag가 add이므로 Matrix Reloaded라는 영화가 추가되었는데 관리 번호는 377, 감독은 Wachowski Brothers, 제작연도는 2003년, 등이라는 것을 알 수 있음
- \* 두 번째 레코드는 tag가 update이므로 관리 번호가 371인 영화의 정보가 수정되었는데 배우 부분이 변경된 것임을 알 수 있음. 즉, 변경되지 않은 항목은 =으로 표시함
- \* 마지막 레코드는 tag가 delete이므로 관리 번호가 354인 영화가 삭제됐음을 알 수 있음

## 2.2. movie\_list 파일

- movie\_log에서 add 항목에 update 및 delete 항목이 적용된 영화 목록. 다음과 같이 저장됨
  - serial\_number:title:genre:director:year:time:actors
- \* 항목의미는 movie\_log와 같음

## 2.3. director\_log 파일

- 감독 목록의 모든 기록을 담고 있는 파일로 다음과 같은 형식으로 저장됨
  - tag:serial\_number:name:sex:birth:best\_movies
- \* tag : 이 레코드의 생성 형태를 알려주는 것으로 add, update, delete가 올 수 있음. 각각은 다음과 같은 의미를 가짐
  - . add : 새로운 감독 정보가 삽입됨
  - . update : 기존의 감독 정보가 수정됨
  - . delete : 기존의 감독 정보가 삭제됨
- \* serial\_number : 감독의 고유 관리번호. 1부터 부여되며 add 될 때마다 숫자가 하나씩 증가함. 삭제된 감독에 할당된 번호는 재사용하지 않음
- \* name : 감독 이름
- \* sex : 성별(M/F)
- \* birth : 생년월일. YYYYMMDD 형식
- \* best\_movies : 대표 작품으로 “,”로 분리되어 들어감
- director\_log 파일의 예는 다음과 같음
  - add:1:Steven Spielberg:M:19461218:Saving Private Ryan
  - add:2:Martin Scorsese:M:19421117:The Wolf of Wall Street
  - add:3:Ridley Scott:M:19371130:Blade Runner
  - update:1:::Saving Private Ryan, Catch Me if You Can, E.T.
  - delete:2:::



#### 2.4. director\_list 파일

- director\_log에서 add 항목에 update 및 delete 항목이 적용된 감독 목록. 다음과 같이 저장됨  
serial\_number:name:sex:birth:best\_movies
- \* 항목의미는 director\_log와 같음

#### 2.5. actor\_log 파일

- 배우 목록의 모든 기록을 담고 있는 파일로 다음과 같은 형식으로 저장됨  
tag:serial\_number:name:sex:birth:best\_movies
- \* tag : 이 레코드의 생성 형태를 알려주는 것으로 add, update, delete가 올 수 있음. 각각은 다음과 같은 의미를 가짐
  - . add : 새로운 배우가 삽입됨
  - . update : 기존의 배우 정보가 수정됨
  - . delete : 기존의 배우 정보가 삭제됨
- \* serial\_number : 배우의 고유 관리번호. 1부터 부여되며 add 될 때마다 숫자가 하나씩 증가함. 삭제된 배우에 할당된 번호는 재사용하지 않음
- \* name : 배우 이름
- \* sex : 성별(M/F)
- \* birth : 생년월일. YYYYMMDD 형식
- \* best\_motives : 대표 작품으로 “,”로 분리되어 들어감
- actor\_log 파일의 예는 다음과 같음  
add:1:Robert De Niro:M:19430817:Goodfellas  
add:2:Jack Nicholson:M:19370422:The Shining  
add:3:Tom Hanks:M:19560709:The Da Vinci Code  
add:4:Marlon Brando:M:19240403:Daeboo  
update:1:=:=:Taxi Driver, The Deer Hunter  
delete:4:::  
add:5:Leonardo DiCaprio:M:19741111:Inception

#### 2.6. actor\_list 파일

- actor\_log에서 add 항목에 update 및 delete 항목이 적용된 배우 목록. 다음과 같이 저장됨  
serial\_number:name:sex:birth:best\_movies
- \* 항목의미는 actor\_log와 같음

#### 2.7. 파일과 관련하여 다음과 같은 요구사항을 만족해야 함

- 1) 모든 파일은 텍스트 파일로 만듦. 즉, 셸에서 파일의 내용을 확인할 수 있어야 함
- 2) 모든 내용은 영어로만 저장함
- 3) 예제에서와 같이 레코드 내 항목 간 구분은 :(콜론)으로 함. 만일 항목 내용에 콜론이 들어간다면 ??;로 표시함. 단 출력할 때에는 콜론으로 출력해야 함
- 4) log 파일은 프로그램이 실행될 때 마다 읽어서 처리해야 함. 즉, log 파일을 읽으면서 update 와 delete tag를 갖는 레코드를 기존 레코드에 반영해야 함
- 5) list 파일은 log 파일 처리가 끝난 직후 파일로 저장하여 만듦. 만일 기존에 list 파일이 있다면 기존의 파일은 현재시간을 확장자로 만들어 복사해두어야 함. 즉, movie\_log 파일의 처리가 끝나면 그 결과를 movie\_list에 저장하는데, 만일 movie\_list 파일이 있다면 기존의 movie\_list를 movie\_list.201711082340 (2017년 11월 8일 23:40)로 복사함

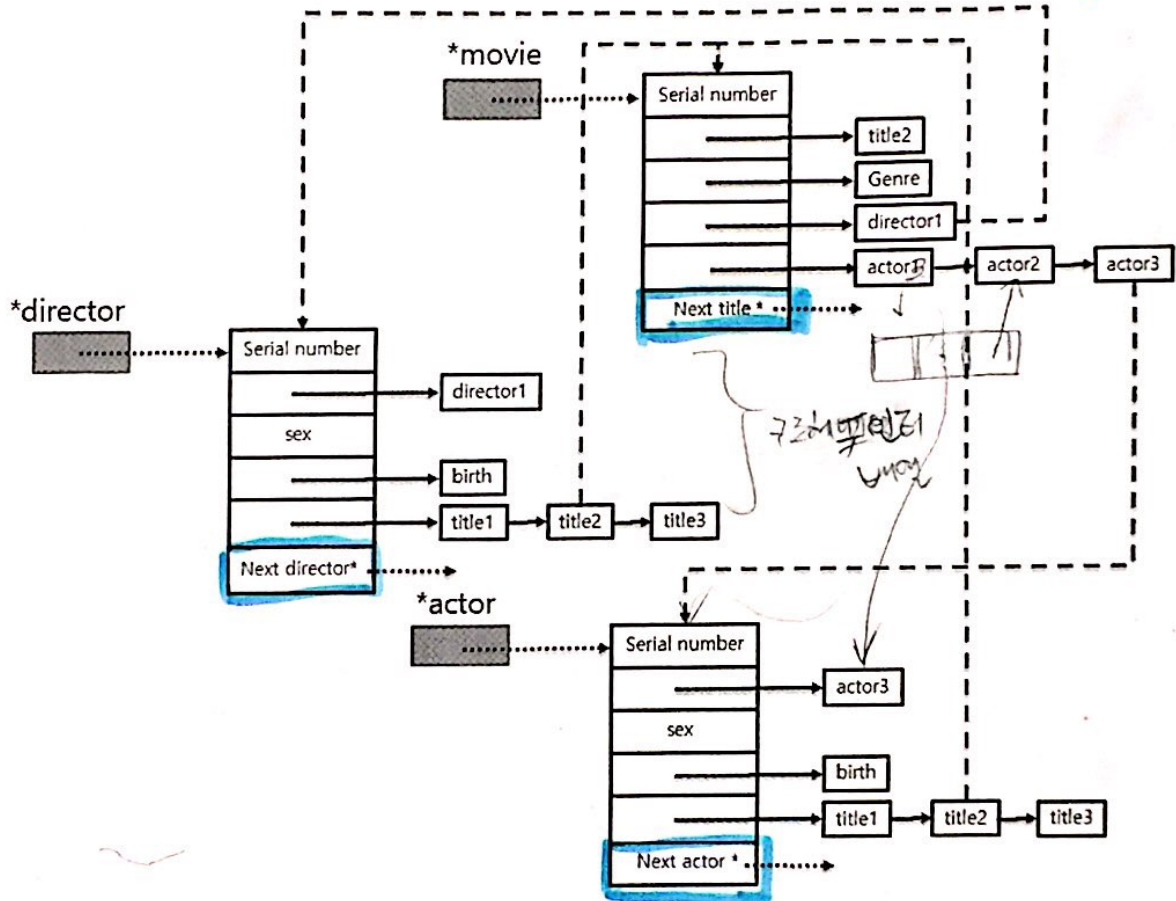
3. 프로그램은 다음 조건을 반영해야 함

3.1. 프로그램은 Cygwin 기본 환경에서 컴파일되고 실행되어야 함

3.2. 프로그램에 배열을 선언하면 안되고 배열 표현을 사용해도 안됨

3.3. 신호(Signal)를 받으면 어떤 신호를 받았는지 보여주고 프로그램 종료 여부를 묻은 후 y(yes)가 입력되면 종료해야 함. 즉, Control-C 신호를 받은 후 바로 종료하면 안됨 ??

3.4. 프로그램이 실행되면 log 파일을 읽어 아래 그림과 같은 구조로 메모리에 저장해야 함



- 영화 목록, 감독 목록, 배우 목록은 각각 movie, director, actor 포인터가 포인트하는 링크드 리스트로 메모리에 저장해야 함
- 영화 레코드의 배우 항목, 감독 레코드의 대표영화 항목, 배우 레코드의 대표영화 항목과 같이 여러 개의 데이터로 이루어진 것은 링크드 리스트 형태로 저장함
- 영화 레코드의 영화감독이 영화감독 리스트에 있다면 영화 레코드 상의 영화감독 항목이 영화감독 리스트에 있는 그 감독 노드를 포인트하게 해야 함. 또한 영화 레코드의 배우가 배우 리스트에 있다면 그 레코드를 포인트하게 해야 함. 이는 감독과 배우 리스트에서도 마찬가지로 적용됨

3.5. 파일 읽기가 끝나면 다음과 같은 프롬프트를 출력하고 사용자의 명령을 기다림

(movie)



3.6. 프로그램은 다음과 같은 명령어를 지원해야 한다.

• search, print, add, update, delete, sort, save, end

#### 3.6.1. search

- 키워드 검색으로 리스트 상에 지정된 문자열을 찾아서 출력함
- 사용법

search [option] string

- string은 검색할 문자열(또는 단어) 임
- string에 메타문자 \*와 ?를 사용할 수 있음. 메타문자 \*는 문자열의 제일 앞과 제일 뒤에만 올 수 있고, 0개 이상의 임의의 문자를 의미함. 즉, st\*는 st로 시작하는 모든 단어를 의미함. ?는 1개의 임의의 문자를 의미함. 즉, st?r는 st로 시작하고 r로 끝나는 4개의 문자로 이루어진 단어를 의미함
- [option]은 검색을 수행할 리스트를 지정하기 위한 것임. 옵션이 없으면 세 개의 리스트에 있는 모든 레코드에서 string과 같은 문자열이 있는지 검색하고 있으면 해당 레코드 전체를 출력함
- [option]으로는 -m, -d, -a를 사용할 수 있고 각각 영화, 감독, 배우 레코드만 검색하라는 의미임. 단, 이 세 가지 옵션은 중복되어 올 수 있는데, 예를 들어 영화와 감독 레코드에서 검색을 원할 경우 -md 또는 -dm으로 지정할 수 있음

#### 3.6.2. print

- 지정된 레코드를 출력함
- 사용법

print m|d|a num

- 인자로 m, d, a 중 하나와 관리번호가 옴. m, d, a는 각각 영화, 감독, 배우 리스트를 의미하고 num은 각 목록의 관리번호를 의미함
- 각 옵션별 출력 형식은 다음과 같음

- 영화 (m)

일련번호, 영화제목, 장르

D : 감독명(생년월일)

A1 : 배우1(생년월일)

A2 : 배우2(생년월일)

...

- 감독 (d)

일련번호, 감독 이름, 성별, 생년월일

대표작1, 대표작1의 제작년도, 상영시간

대표작2, 대표작2의 제작년도, 상영시간

...

- 배우 (a)

일련번호, 배우 이름, 성별, 생년월일

대표작1, 대표작1의 제작년도, 상영시간

대표작2, 대표작2의 제작년도, 상영시간

...

- 영화 레코드를 출력할 때 감독이나 배우 관련 세부 정보, 감독 레코드를 출력할 때 영화 관련 세부 정보, 배우 레코드를 출력할 때 영화 관련 세부정보는 메모리에 유지하고 있는 링크를 통해 출력해

야 함

### 3.6.3. add

- 새로운 레코드를 삽입함
- 사용법

add m|d|a

- 인자로 m, d, a 중 하나가 올 수 있고 각각 영화, 감독, 배우 레코드를 삽입한다는 의미함
- 입력 내용은 대화식으로 받음, 즉, 영화 레코드를 입력받을 경우 다음과 같이 입력할 항목 이름을 출력하여 사용자가 알 수 있게 함

(movie) add m

title > XXX

genre > action

director > ....

....

(movie)

- 입력된 레코드와 같은 레코드(영화인 경우 제목, 감독과 배우인 경우 이름)가 이미 존재하면 "You have the same record." 라는 메시지를 출력하고, 기존 레코드를 출력해 준 후 입력한 내용을 새로운 레코드로 추가할지 다시 물어본 후 추가여부를 결정함
- 레코드가 성공적으로 삽입되면, 관련 log 파일에 그 내용을 삽입하고 메모리 정보도 적절히 수정해야 함. 즉, 영화 레코드가 삽입됐다면 movie\_log 파일에 add 태그를 갖는 레코드가 하나 삽입되어야 하고, 메모리의 영화 리스트의 제일 마지막에 삽입된 레코드를 삽입하고 (필요시 director, actor 포인터도 삽입해야 하며, 감독 및 배우 리스트의 영화관련 링크도 수정해야 함)

### 3.6.4. update

- 기존 레코드를 수정함
- 사용법

update m|d|a [option] num

- 인자로 m, d, a 중 하나와 num(관리번호, serial number)가 올. m, d, a는 각각 영화, 감독, 배우 리스트를 의미함
- [option]은 지정된 레코드의 특정 필드만을 수정할 때 지정하는 것으로 다음과 같은 문자를 사용하고 중복해서 올 수 있음

영화 레코드(m)		감독 레코드(d)		배우 레코드(a)	
t	제목	n	이름	n	이름
g	장르	s	성별	s	성별
d	감독	b	생년월일	b	생년월일
y	년도	m	영화	m	영화
r	시간				
a	배우				

- 예를 들어 영화 목록에서 관리번호가 14인 영화 레코드의 감독이름과 상여시간을 수정하고 싶으면 다음과 같이 함

(movie) update m dr 14



- 입력되는 내용은 대화식으로 받음. 즉, 다음과 같이 입력받을 항목 명을 출력해 줘야 함  
(movie) update m dr 14

director > James

running time > 125

(movie)

- 수정된 레코드와 같은 레코드(영화인 경우 제목, 감독과 배우인 경우 이름)가 이미 존재하면 "You have the same record." 라는 메시지를 출력하고, 기존 레코드를 출력해 준 후 수정한 내용을 반영할지 다시 물어본 후 수정여부를 결정함
- 레코드가 성공적으로 수정되면, 관련 log 파일에 그 내용을 삽입하고 메모리 정보도 적절히 수정해야 함. 즉, 영화 레코드가 수정됐다면 movie\_log 파일에 update 태그를 갖는 레코드가 하나 삽입되어야 하고, 메모리의 영화 리스트 상의 적절한 노드 정보를 수정하고, 필요시 director, actor 리스트에서 그 노드를 포인팅 하고 있는 포인터도 수정해야 함

### 3.6.5. delete

- 기존 레코드를 삭제함
- 사용법

delete m|d|a num

- 인자로 m, d, a 중 하나와 num(관리번호, serial number)이 옴. m, d, a는 각각 영화, 감독, 배우 리스트를 의미함
- 관리번호가 num인 레코드가 없다면 "No such record"라는 메시지를 출력하고 종료함
- 그런 레코드가 있다면 리스트에서 그 노드를 삭제해야 함
- 레코드가 성공적으로 삭제되면, 관련 log 파일에 그 내용을 삽입하고 메모리 정보도 적절히 수정해야 함. 즉, 영화 레코드가 삭제됐다면 movie\_log 파일에 delete 태그를 갖는 레코드가 하나 삽입되어야 하고, 메모리의 영화 리스트 상의 적절한 노드 정보를 삭제해야 하며, 필요시 director, actor 리스트에서 그 노드를 포인팅 했던 노드들의 정보도 수정해야 함

### 3.6.6. sort

- 목록을 정렬하여 출력함
- 사용법

sort m|d|a [option] [-f file\_name]

- 인자로 m, d, a 중 하나가 올 수 있고 각각 영화, 감독, 배우 리스트를 의미함
- option이 지정되지 않다면 영화, 감독, 배우 레코드를 각각 (영화제목, 감독이름, 배우이름)으로 정렬하여 출력함
- option은 지정된 레코드의 특정 필드를 기준으로 정렬하라는 의미이고, 다음과 같은 문자를 사용함

영화 레코드(m)		감독 레코드(d)		배우 레코드(a)	
t	제목	n	이름	n	이름
g	장르	s	성별	s	성별
d	감독	b	생년월일	b	생년월일
y	년도	m	영화	m	영화
r	시간				
a	배우				

- 만일 file\_name이 지정되어 있다면 지정된 파일에 저장함

- 정렬은 메모리에 유지하고 있는 링크드 리스트를 사용하지 않고 movie\_list, director\_list, actor\_list 파일을 읽어 포인터 배열을 사용하여 수행해도 됨

### 3.6.7. save

- 리스트 내용을 파일로 저장함
- 사용법

save m|d|a [option] [-f file\_name]

- 인자로 m, d, a 중 하나가 올 수 있고 각각 영화, 감독, 배우 리스트를 의미함
- option과 file\_name이 지정되지 않다면 디폴트 파일에 저장하는데 각 목록 별 디폴트 파일 명은 다음과 같음

구분	파일명
영화	movie_list
감독	director_list
배우	actor_list

- 만일 file\_name이 지정되어 있다면 지정된 파일에 저장함
- [option]은 지정된 목록의 특정 필드만을 지정하여 저장할 때 사용하는 것으로 다음과 같은 문자를 사용함

영화 레코드(m)		감독 레코드(d)		배우 레코드(a)	
t	제목	n	이름	n	이름
g	장르	s	성별	s	성별
d	감독	b	생년월일	b	생년월일
y	년도	m	영화	m	영화
r	시간				
a	배우				

- 즉, 다음과 같은 명령은 영화 레코드 중 제목, 감독, 년도만을 my\_list에 저장하라는 의미이다.

save m -tdy -f my\_list

### 3.6.8. end

- 프로그램을 종료함
- 필요하다면 프로그램 종료 전에 모든 정보를 파일로 저장해야 함

종료?



#### 4. 화면 구성 예

```
$ myMOVIE
>> Welcome to My Movie <<
File Loading.....
You can use add, update, delete, search, sort, save, end commands.
(movie)
```

```
(movie) add m Enter
title > XXX Enter
genre > action Enter
director > Rob Cohen Enter
year > 2002 Enter
run time > 124 Enter
actors > Vin Diesel, Samuel L. Jackson, Marton Csokas, Asia Argento,
Michael Roof, Eve Enter
@@ Done

(movie) add m Enter
title > XXX Enter
genre > action Enter
director > Rob Cohen Enter
year > 2002 Enter
run time > 124 Enter
actors > Vin Diesel, Samuel L. Jackson, Marton Csokas, Asia Argento,
Michael Roof, Eve Enter
@@ You have the same record in movie list.
298:XXX:action:Rob Cohen:2002:124:Vin Diesel, Samuel L. Jackson,
Marton Csokas, Asia Argento, Michael Roof, Eve
@@ Do you want to add any way? No Enter

(movie)
```

```
(movie) update m tg 298[Enter]
title > xXx[Enter]
genre > Action[Enter]
@@ Done
(movie) search xXx[Enter]
298:xXx:Action:Rob Cohen:2002:124:Vin Diesel, Samuel L. Jackson,
Marton Csokas, Asia Argento, Michael Roof, Eve
@@ Done

(movie) delete m 298[Enter]
@@ Done

(movie) delete m 298[Enter]
@@ No such record.

(movie) search xXx[Enter]
@@ No such record.

(movie)
```

```
(movie) print m 377[Enter]
377, Matrix Reloaded, Sci-Fi
  D :Wachowski Brothers(19671229)
  A1:K. Reeves(19641102)
  A2:Fishburne(-)
  A3:C.A. Moss(19670821)
  . . .

(movie)
```



```
(movie) sort m g -f movie_genre 
```

```
Control+C
```

```
Get Interrupt Signal.
```

```
Do you want to exit myMOVIE program? (Y/N) N 
```

```
(movie)
```

```
(movie) end
```

```
$
```