Tutorial 6

NWEN 241
Systems Programming

Alvin C. Valera

alvin.valera@ecs.vuw.ac.nz

Content

- Dynamic memory allocation in C and C++
- STL Vector

Tutorial Problem

You are given data about the greatest sci-fi and fantasy films of all time (taken from https://docs.google.com/spreadsheets/d/1okTb4MTllkDWlj0daHNfA8RwOPhLQkWtpQN_znJJJmk/edit?hl=en&hl=en#gid=5):

- Define a structure that can hold a record
- Allocate enough memory based on user input
- Write a program to allow user to input records and store in the allocated memory and display

Data

Title	Director	Year of Release	Oscars Won	Country
2001	Stanley Kubrick	1968	1	USA
Metropolis	Fritz Lang	1927	0	Germany
Blade Runner	Ridley Scott	1982	0	USA
Alien	Ridley Scott	1979	1	USA
The Wizard of Oz	Victor Fleming	1939	2	USA
ET	Steven Spielberg	1982	4	USA
Solaris	Andrei Tarkovsky	1972	0	USA
Spirited Away	Hayao Miyazaki	2001	1	Japan
Star Wars (1977)	George Lucas	1977	6	USA
Close Encounters	Steven Spielberg	1977	1	USA
King Kong	Ernest B Schoedsack; Merian C Cooper	1933	0	USA
Terminator/Terminator 2	James Cameron	1984	4	USA
The Matrix	Andy & Larry Wachowski	1999	4	USA
Alphaville	Jean Luc-Godard	1965	0	France
Back to the Future	Robert Zemeckis	1985	1	USA
Planet of the Apes	Franklin J Schaffner	1968	1	USA
Brazil	Terry Gilliam	1985	0	UK
The Lord of the Rings trilogy	Peter Jackson	2001	17	New Zealand
Dark Star	John Carpenter	1974	0	USA
Day the Earth Stood Still	Robert Wise	1951	0	USA
Edward Scissorhands	Tim Burton	1990	0	USA
Akira	Katsuhiro Otomo	1988	0	Japan
Princess Bride	Rob reiner	1987	0	USA
Pan’s Labyrinth	Guillermo del Toro	2006	3	Spain
Starship Troopers	Paul Verhoeven	1997	0	USA

Define a structure that can hold a record

```
#define DEFAULT_STRLEN      100

struct movie {
      char title[DEFAULT_STRLEN];
      char director[DEFAULT_STRLEN];
      short year;
      short oscars_won;
      char origin_country[DEFAULT_STRLEN];
};

typedef struct movie movie_t;
```

Allocate enough memory to hold records

Approach #1: Use static array

```
int size;

// Ask user to input size

movie_t movies[size];
```

Putting it all together

- Write a program to allow user to input records and store in the allocated memory and display
  - See `t6a.c`

- What if more data needs to be entered later?

Allocate enough memory to hold records

Approach #2: Use `malloc` or `calloc` to allocate dynamic memory

```
int size;

// Ask user to input size

movie_t *movies = (movie_t *)
     calloc(size, sizeof(movie_t));
```

```
movie_t *movies = (movie_t *)
     malloc(size*sizeof(movie_t));
```

Putting it all together

- Write a program to allow user to input records and store in the allocated memory and display
  - See `t6b.c`

- What if more data needs to be entered later?
  - Use `realloc()` to allocate more memory
  - See `t6b2.c`

- What if a record needs to removed?
  - Allocate new memory and copy data (excluding removed record)
  - See `t6b3.c`

C++

Allocate enough memory to hold records

Approach #3: Use `vector` from C++ standard template library

```
int size;

// Ask user to input size

// vector does not require specification of size
// beforehand
vector<movie_t> movies;
```

Putting it all together

- Write a program to allow user to input records and store in the allocated memory and display
  - See `t6c.c`

- What if more data needs to be entered later?
  - Just call `push_back()` member function to add more items into vector
  - See `t6c2.c`

- What if a record needs to removed?
  - Just call `erase()` member function to remove item from vector
  - See `t6c3.c`