

CS-240

...

1st Programming Project

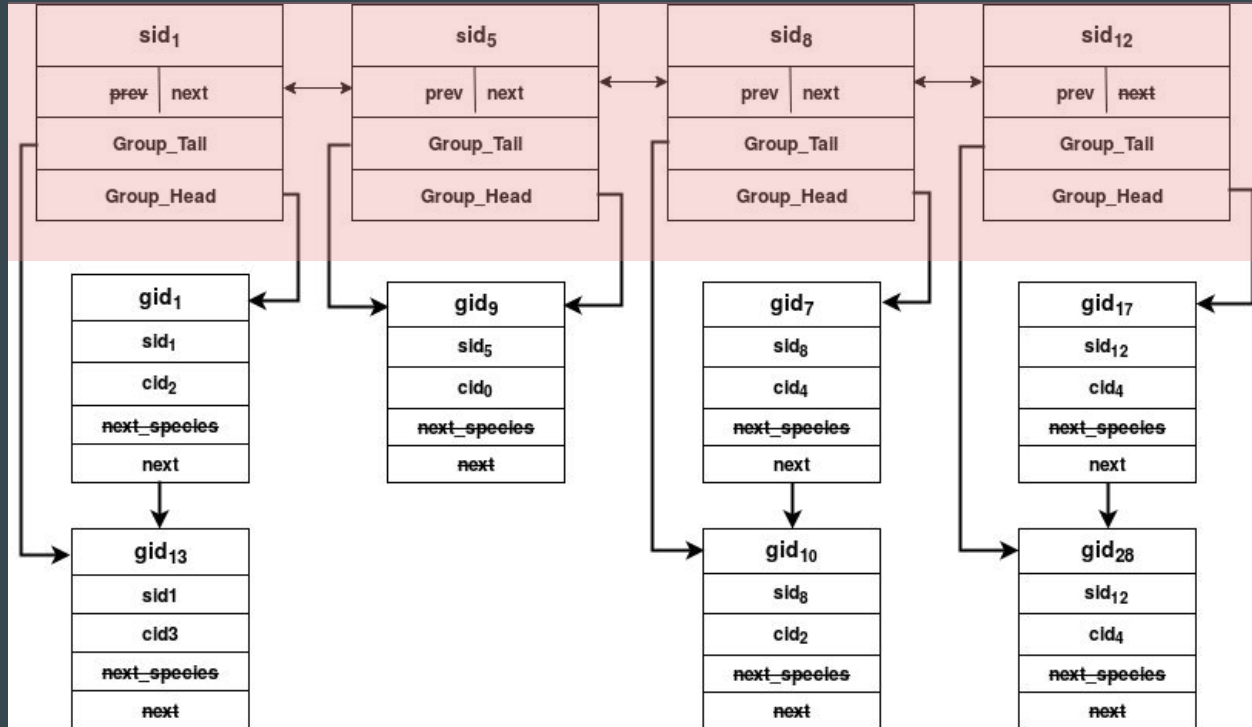
Project Outline & Entities

- ❖ Simulate human species evolution
- ❖ Different **species**
- ❖ Each species consists of different **populations**
- ❖ Each population resides in a **continent**

Species

- ❖ Species are stored in a *double-linked* list
- ❖ **Sorted** on species' ID
- ❖ struct *Species*
 - int sid; // *species' ID*
 - struct Species *next;
 - struct Species *prev;
 - struct Population *head;
 - struct Population *tail;

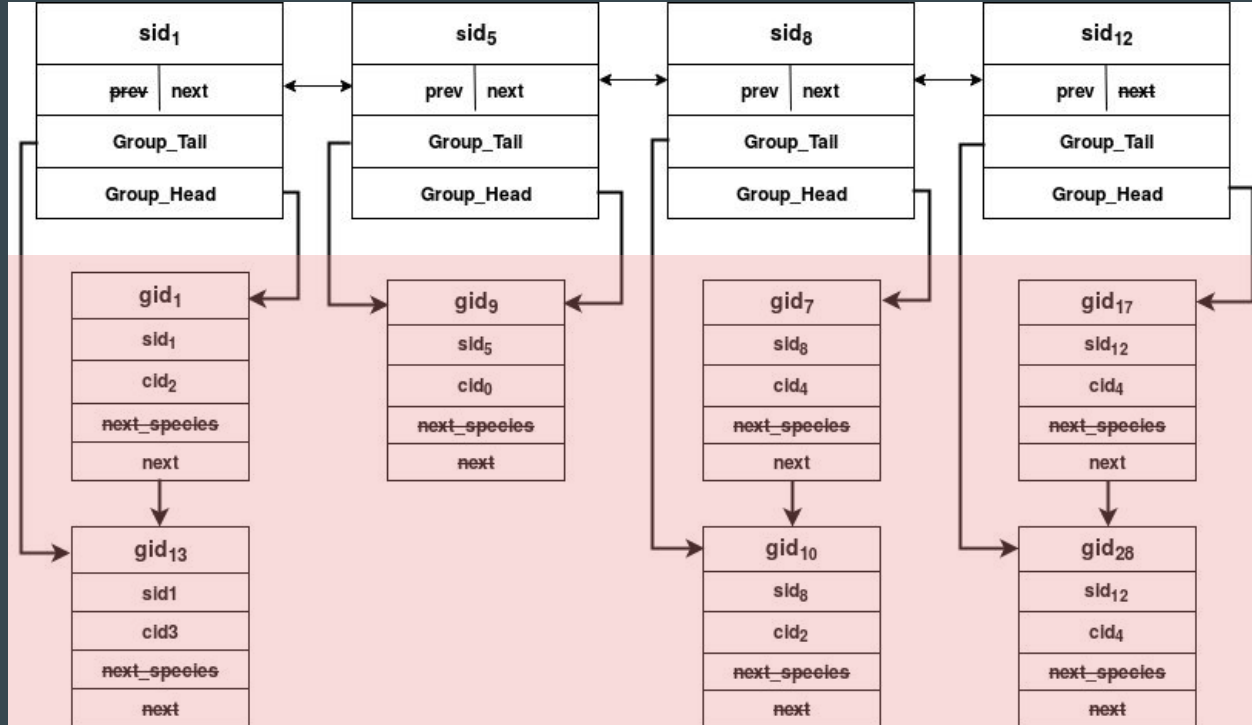
Species Double-linked List



Populations

- ❖ Populations are stored in a **single-linked list**
- ❖ **Sorted** on population's ID
- ❖ struct Population
 - int gid; // *population ID*
 - int sid; // *species' ID*
 - int cid; // *continent's ID*
 - struct Population *next;
 - struct Population *next_species;

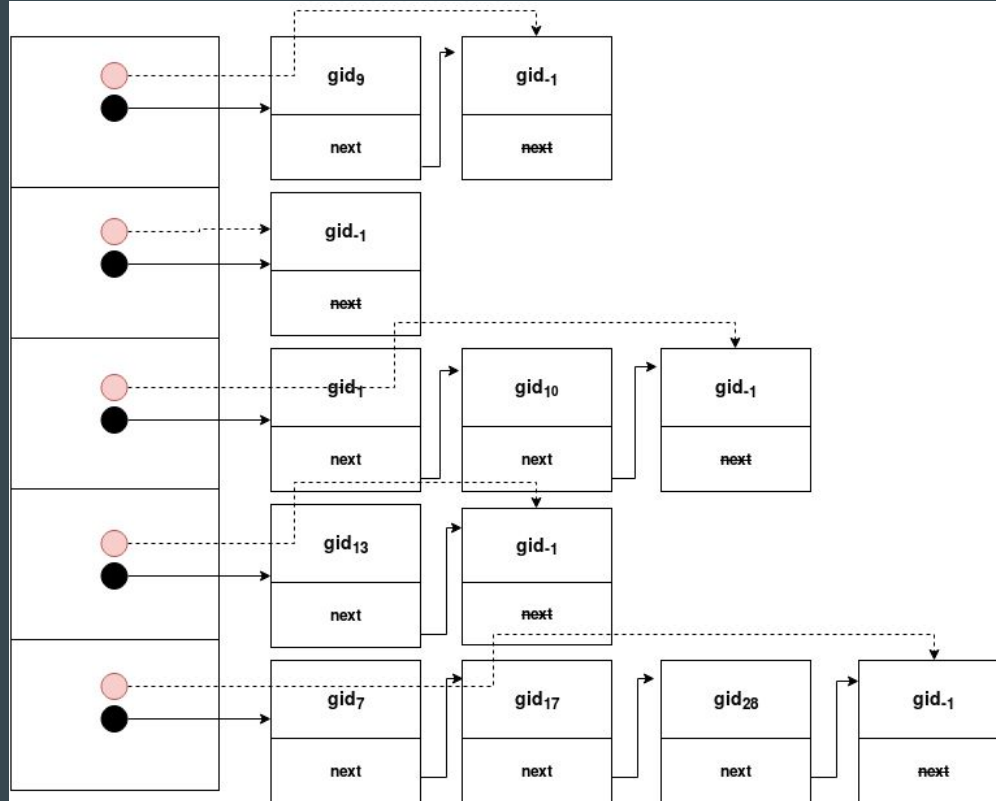
Species Double-linked List & Populations



Continents

- ❖ Continents are described by an **array of size 5**
- ❖ Each element is a pointer to a *struct ContinentList*, containing the information
- ❖ The lists are ***unsorted*** and have a ***sentinel*** node
- ❖ struct ContinentPopulation
 - int gid; // *population's ID*
 - struct ContinentPopulation *next;

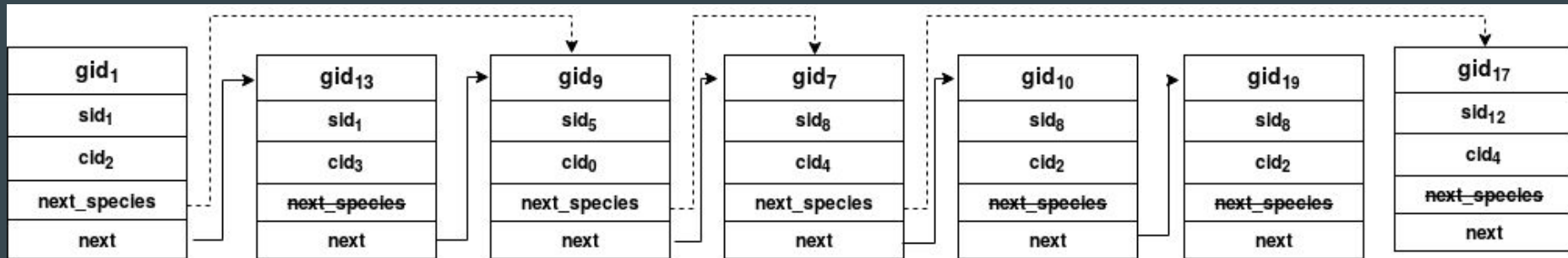
Continent's Array



Homo Sapiens

- ❖ All species eventually evolve (or contribute) to *Homo Sapiens*
- ❖ The evolution is described by a ***single-linked list, sorted*** on the species' ID
- ❖ Each node is of type ***struct Population***
- ❖ Each species' 1st node points to the 1st population of the next species (*next_species* pointer)

Homo Sapiens List



Events

- ❖ **S <sid>** (new species)
 - Insert new species node in species DLL
- ❖ **G <gid> <sid> <cid>** (new population)
 - Traverse species list and locate <sid>
 - Insert new node in its population list
- ❖ **M <sid1> <sid2> <sid3>** (merge species)
 - Traverse species list and locate <sid1> and <sid2>
 - Insert new species with <sid3>
 - **Transfer** <sid1> and <sid2> populations, in <sid3> (complexity $O(n+m)$)
 - New population list should still be sorted on the populations' ID
 - Delete <sid1> and <sid2>

More events

❖ D (distribute)

- Traverse **whole** population list for each species
- For each population, insert a new node (*struct ContinentPopulation*) in its continent's list
- Use the population's **<cid>** field to locate the correct continent

❖ K <gid> <sid> (delete population)

- Traverse species list and locate **<sid>**
- Delete population **<gid>**
- Delete population from its continent as well

More events

❖ F <sid> (species extinction)

- Traverse species list in **descending** order
- Locate and delete species <sid>
- Delete all of its populations
- Delete each population from the continent's array as well

❖ E (evolution)

- Traverse species list
- **Transfer** each population list **as-is** to the evolution list
- Update the **next_species** pointer of a species' 1st population to point to the 1st of the next one
- **O(n)** : n is the number of species

More events

❖ N (species statistics)

- Traverse the **homo Sapiens** list and count all species
- $O(n)$: n is the number of species

❖ J <sid> (population statistics)

- Traverse **homo Sapiens** list and locate the first population of <sid>
- Traverse that specific species' population list and count its populations
- $O(n+m)$: n is the number of species; m is the number of populations that belong to <sid>

Print events

- ❖ **P**: Print species list, **without** their population lists
- ❖ **X**: Print species list, **with** population lists
- ❖ **C**: Print continent's array and each continent's population list
- ❖ **W**: Print homo Sapiens list

General Info

- ❖ Your project should:
 - Contain everything it needs to compile
 - Compile & run in CSD's UNIX machines
- ❖ For any questions, contact the TAs using the course's e-mail or mailing list
 - hy240a@csd.uoc.gr
 - hy240a-list@csd.uoc.gr