Mobile Computing
Lecture 26
Palm OS 2

Contents

- System managers
- Data manager
- Resource manager
- Application structure

The Memory Manager

- The functions of the memory manager:
  - allocating new chunks
  - disposing of chunks
  - resizing of chunks
  - locking and unlocking chunks
  - compacting heaps when they become fragmented

Heap Structure

- Master pointer table stores 32-bit pointer to movable chunks
- Movable chunks are allocated at the beginning

Heap
Heap Header
Master pointer table
Movable chunks
Non-movable chunks
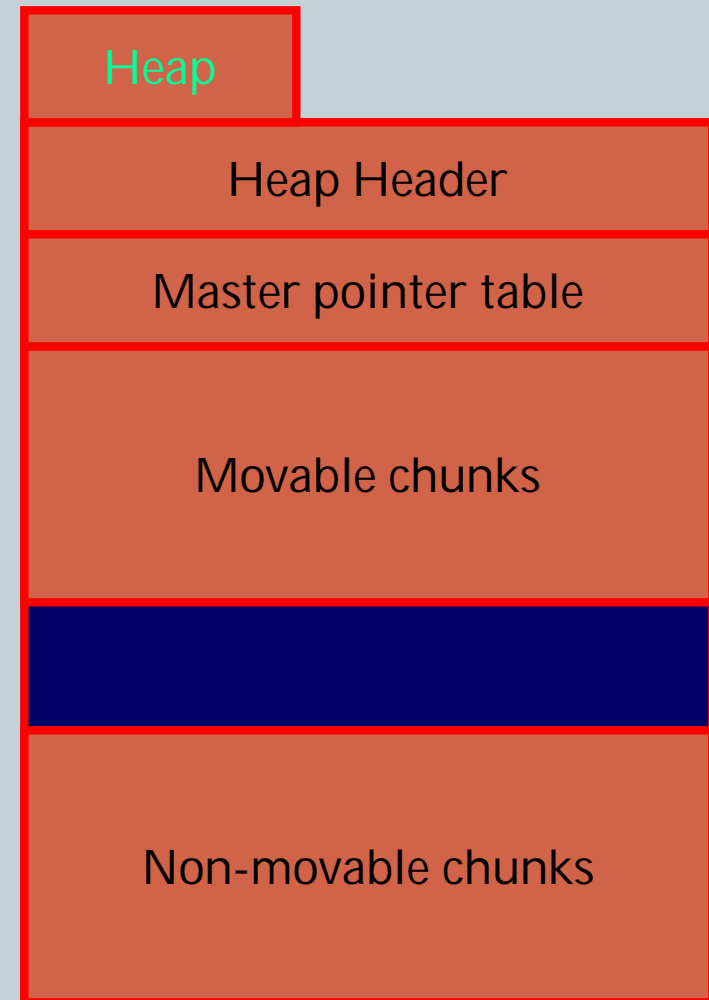
Chunk Structure

- Each chunk begins with an 8-byte header followed by that chunk's data

- Flags:sizeAdj flag:
  - high nibble : set for free chunk
  - low nibble : reqSize = size - 8 - [this value]

- size field(3 bytes)
  - the size of the chunk, which is larger than the size requested by the application , including the chunk header itself

Chunk Structure(cont.)

- Lock:owner byte
  - high nibble : the lock count, which is incremented when being locked
  - low nibble : the owner ID of the memory chunk
- hOffset field(3 bytes)
  - the distance from the master pointer to the chunk header, divided by two

The Data Manager

- The database is analogous to disk

- A database is a collection of records

- A record is mapped to a memory chunk

- A database accesses its records by storing their **local ID's**

- An application requests a particular record in a database by index

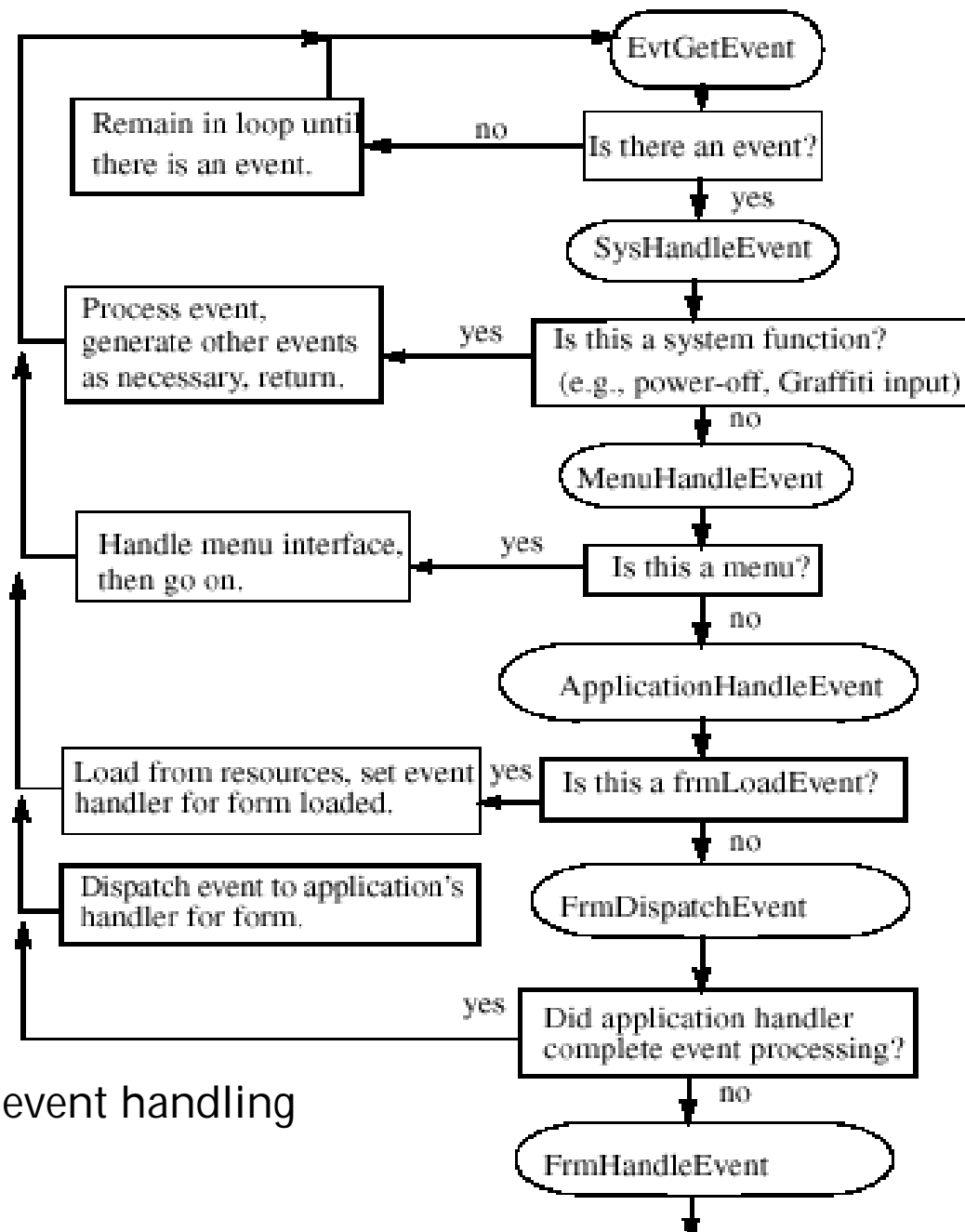# The Resource Manager

- Resources : store the UI elements of an application, such as images, fonts, dialog layouts, …

- Resource manager : Data manager with the additional ability of tagging each chunk of data with a unique resource type and resource ID

Application Structure

- Single threaded

- Event-driven

- PilotMain() corresponds to main() in C

- PilotMain -- Response to launch codes

- An event loop in response of a normal launch

EvtGetEvent

Remain in loop until there is an event. ← no ← Is there an event?

yes

SysHandleEvent

Process event, generate other events as necessary, return. ← yes ← Is this a system function? (e.g., power-off, Graffiti input)

no

MenuHandleEvent

Handle menu interface, then go on. ← yes ← Is this a menu?

no

ApplicationHandleEvent

Load from resources, set event handler for form loaded. ← yes ← Is this a frmLoadEvent?

no

FrmDispatchEvent

Dispatch event to application's handler for form. ← yes ← Did application handler complete event processing?

no

FrmHandleEvent

Flowchart of event handling

Developing Apps on Palm

- Various development tools:
  - Code Warrior for Palm OS
  - Palm SDK (header files, documents, examples)
  - GCC
  - POSE - An open source Palm emulator runs on Windows, Mac and UNIX

Conclusion

- The Palm OS provides a good platform for developing Palm apps
- The Palm OS provides various libraries for communicating with PC's
- A good OS design is not necessarily having the most advanced feature, but having the best integration of the hardware