

## ບົດທີ 2

ແນວຄິດກ່ຽວກັບ Hardware ແລະ  
Software

# ເນື້ອໃນຫຍໍ້

- ◆ ສະເໜີເບື້ອງຕົ້ນ
- ◆ ວິວັດທະນາການຂອງອຸປະກອນ Hardware
- ◆ ອົງປະກອບຂອງ Hardware
- ◆ Hardware ທີ່ສະໜັບສະໜູນລະບົບປະຕິບັດການ
- ◆ Caching ແລະ Buffering
- ◆ ນຳສະເໜີກ່ຽວກັບ Software
- ◆ Application Programming Interface (API)
- ◆ Compiling, Linking ແລະ Loading
- ◆ Firmware
- ◆ Middleware

# ສະເໜີເບື້ອງຕົ້ນ

- ◆ ລະບົບປະຕິບັດການເປັນຜູ້ບໍລິຫານຈັດການຊັບພະຍາກອນຫຼັກໃນລະບົບຄອມພິວເຕີ
- ◆ ໄດ້ຖືກອອກແບບມາເພື່ອໃຫ້ເຮັດວຽກກັບຮາດແວຣ໌ ແລະ ຊອບແວຣ໌ທີ່ມັນບໍລິຫານ ຊຶ່ງມີດັ່ງນີ້
  - ◆ processors
  - ◆ memory
  - ◆ secondary storage (such as hard disks)
  - ◆ other I/O devices
  - ◆ processes
  - ◆ threads
  - ◆ files and databases

# ວິວັດທະນາການຂອງອຸປະກອນ Hardware

- ❑ ໃນເບື້ອງຕົ້ນ ຮາດແວຣ໌ຍັງມີຈຳນວນໜ້ອຍ ແລະ ບໍ່ທັນສະໄໝ ລະບົບປະຕິບັດການຈຶ່ງເປັນການຂຽນລະຫັດເພື່ອບໍລິຫານຈັດການຮາດແວຣ໌
- ❑ ຕໍ່ມາ ຮາດແວຣ໌ ໄດ້ຖືກພັດທະນາຢ່າງກ້າວກະໂດດ ເຮັດໃຫ້ລະບົບປະຕິບັດການສະຫຼັບຊັບຊ້ອນຂຶ້ນ
- ❑ ເພື່ອອຳນວຍຄວາມສະດວກໃຫ້ແກ່ການຂຽນໂປຣແກຣມຕິດຕໍ່ກັບລະບົບ ແລະ ໃຫ້ລະບົບເຮັດວຽກກັບອຸປະກອນຫຼາຍຢ່າງຂຶ້ນ ລະບົບປະຕິບັດການຈຶ່ງຖືກສ້າງໃຫ້ເປັນອິດສະລະກັບການກຳໜົດຄ່າຂອງຮາດແວຣ໌
- ❑ ອຸປະກອນທຸກຢ່າງຈະຕ້ອງມີ driver ເປັນຂອງຕົວເອງ ແລະ ຮາດແວຣ໌ຫຼາຍຢ່າງຖືກອອກແບບມາໃຫ້ເຮັດວຽກກັບລະບົບປະຕິບັດການ

# ອົງປະກອບຂອງ Hardware

- ◆ ຮາດແວຣ໌ຂອງຄອມພິວເຕີປະກອບດ້ວຍ
  - ໜ່ວຍປະມວນຜົນ (processor(s))
  - ໜ່ວຍຄວາມຈຳຫຼັກ (main memory)
  - ໜ່ວຍປ້ອນຂໍ້ມູນ/ສະແດງຜົນ (input/output devices)

# ອົງປະກອບຂອງ Hardware

## ◆ ແຜງວົງຈອນຫຼັກ (Mainboard)

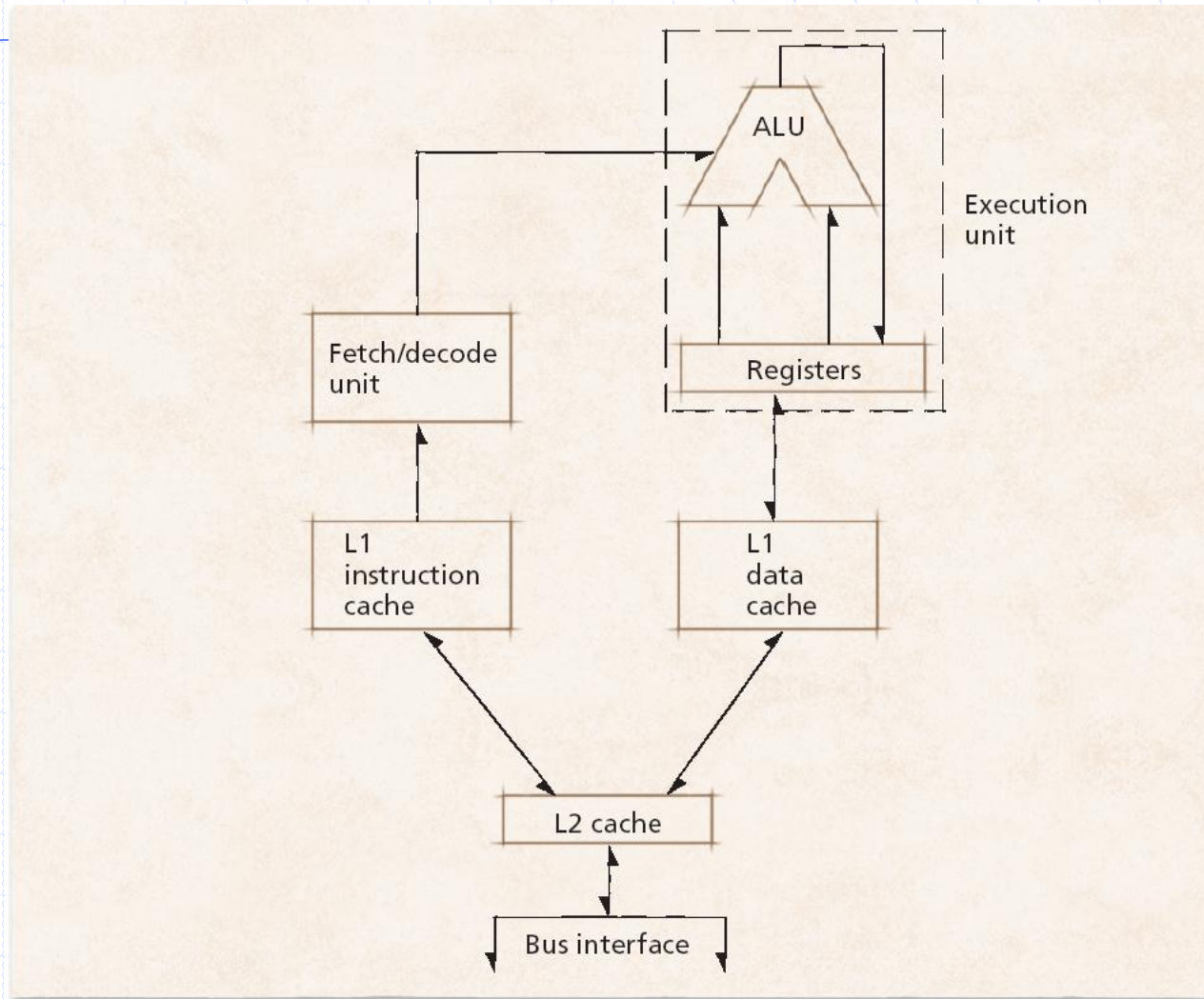
- ເປັນອຸປະກອນຮາດແວຣ໌ຊຶ່ງເປັນບ່ອນເອົາບັນດາອຸກອນຕ່າງຂອງຄອມພິວເຕີມາເຊື່ອມຕໍ່ເຂົ້າກັນ
- ມັນເປັນສູນລວມວົງຈອນເອເລັກໂທຣນິກໃນລະບົບທີ່ໃຊ້ເຊື່ອມຕໍ່ບັນດາອຸປະກອນຕ່າງໆ
  - ◆ ເປັນບ່ອນສຽບໜ່ວຍປະມວນຜົນ ແລະ ຄວາມຈໍາຫຼັກ
  - ◆ ເປັນບ່ອນສຽບບັນດາ chip ຕ່າງໆທີ່ໃຊ້ເຮັດວຽກພື້ນຖານເຊັ່ນ: BIOS

# ອົງປະກອບຂອງ Hardware

## ◆ ໜ່ວຍປະມວນຜົນກາງ (Processor)

- ເປັນອຸປະກອນຮາດແວທີ່ໃຊ້ດໍາເນີນການກັບພາສາເຄື່ອງຄອມພິວເຕີ
- CPU ດໍາເນີນການກັບບັນດາຄໍາສັ່ງຂອງໂປຣແກຣມ
- ນອກຈາກນັ້ນຍັງມີໜ່ວຍປະມວນຜົນເຮັດວຽກພິເສດບາງຢ່າງເຊັ່ນ:  
ປະມວນຜົນຮູບພາບ, ປະມວນຜົນກ່ຽວກັບສຽງ
- ຢູ່ທາງໃນໜ່ວຍປະມວນຜົນປະກອບດ້ວຍ Registers ຈຳນວນໜຶ່ງ  
ຊຶ່ງເປັນໜ່ວຍຄວາມຈໍາຄວາມໄວສູງ
  - ◆ ຂໍ້ມູນຈະຕ້ອງເອົາມາເກັບໄວ້ໃນ Registers ກ່ອນໜ່ວຍປະມວນຜົນຈະ  
ຈັດການມັນໄດ້
- ຄວາມຍາວຂອງຄໍາສັ່ງເປັນຂະໜາດຂອງຄໍາສັ່ງພາສາເຄື່ອງທີ່ໜ່ວຍ  
ປະມວນຜົນສາມາດປະມວນໄດ້
  - ◆ ໜ່ວຍປະມວນຜົນບາງລຸ້ນສາມາດປະມວນຜົນໄດ້ຫຼາຍຂະໜາດ

# ອົງປະກອບຂອງ Hardware





# ອົງປະກອບຂອງ Hardware

## ◆ ສັນຍານໂມງ (Clock)

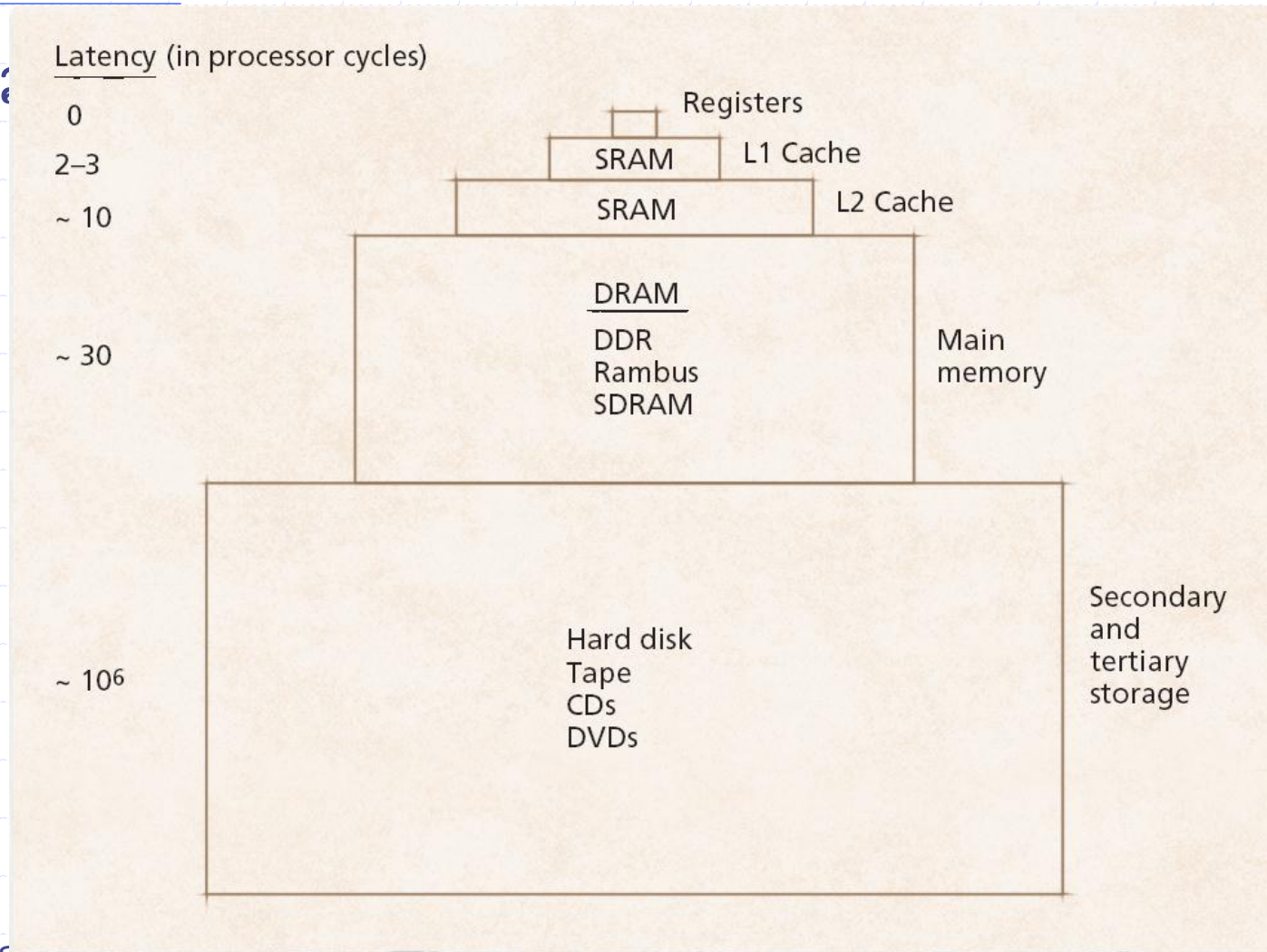
- ເວລາຂອງຄອມພິວເຕີຈະວັດແທກເປັນຮອບວຽນ, ແຕ່ລະຮອບວຽນແມ່ນນັບຕາມການສັ່ນໄກວຂອງສັນຍານໄຟຟ້າຊຶ່ງສ້າງໂດຍວົງຈອນສ້າງສັນຍານໂມງຂອງລະບົບ (System clock generator)
- ຄວາມໄວຂອງໜ່ວຍປະມວນຜົນຈະຖືກວັດແທກເປັນຈິກກະເຮັດ (GHz) ຊຶ່ງແມ່ນໜຶ່ງພັນລ້ານຮອບຕໍ່ວິນາທີ
  - ◆ ຄອມພິວເຕີສະໃໝໃໝ່ສາມາດປະມວນຜົນດ້ວຍຄວາມໄວຫຼາຍຈິກກະເຮັດ

# ອົງປະກອບຂອງ Hardware

## ◆ ລຳດັບຊັ້ນຂອງໜ່ວຍຄວາມຈຳ(Memory Hierarchy)

- ເປັນຮູບແບບໃນການຈັດປະເພດຂອງໜ່ວຍຄວາມຈຳຈາກຂະໜາດນ້ອຍໄປຫາຂະໜາດໃຫຍ່, ຄວາມໄວສູງໄປຫາຄວາມໄວຕໍ່າ ແລະ ລາຄາແພງໄປຫາລາຄາຖືກ ດັ່ງນີ້
  - ◆ Registers
  - ◆ L1 Cache
  - ◆ L2 Cache
  - ◆ Main Memory
  - ◆ Secondary and tertiary storage (CDs, DVDs and floppy disks)
- ໜ່ວຍປະມວນຜົນຈະບໍ່ຄ່ອຍເອີ້ນຫາຂໍ້ມູນຈາກໜ່ວຍຄວາມຈຳຫຼັກໂດຍກົງ ຊຶ່ງມັນເກັບຂໍ້ມູນໄດ້ຊື່ວ່າງເທົ່ານັ້ນ

# ອົງປະກອບຂອງ Hardware



# ອົງປະກອບຂອງ Hardware

## ◆ ໜ່ວຍຄວາມຈຳຫຼັກ(Main Memory)

- ປະກອບດ້ວຍ RAM
- ໂດຍປົກກະຕິ RAM ຈະມີ 2 ຮູບແບບ
  - ◆ dynamic RAM (DRAM) – ຕ້ອງໄດ້ refresh ວົງຈອນ
  - ◆ static RAM (SRAM) – ບໍ່ຕ້ອງ refresh ວົງຈອນ
- ເປົ້າໝາຍຫຼັກຂອງ DRAM ແມ່ນເພື່ອໃຫ້ຄວາມໄວຂອງມັນໄກ້ຄຽງກັບຄວາມໄວຂອງ CPU ຊຶ່ງອອກແບບໃຫ້ລຸດເວລາໃນການເຂົ້າຫາຂໍ້ມູນ ແລະ ເພີ່ມຈຳນວນຄັ້ງໃນການຂົນສົ່ງຂໍ້ມູນໃນເວລາໜຶ່ງວິນາທີ
- Bandwidth ແມ່ນປະລິມານຂໍ້ມູນທີ່ສາມາດສົ່ງໃນຫົວໜ່ວຍເວລາໃດໜຶ່ງ

# ອົງປະກອບຂອງ Hardware

- ◆ **ໜ່ວຍຄວາມຈຳສຳຮອງ (Secondary Memory)**
  - ເປັນໜ່ວຍຄວາມຈຳທີ່ສາມາດເກັບຂໍ້ມູນໄດ້ໃນປະລິມານທີ່ຫຼາຍຢ່າງ ຖາວອນໄດ້ ແລະ ລາຄາຖືກ
  - ການເຂົ້າຫາຂໍ້ມູນຈະຊຶ່ງກ່ວາໜ່ວຍຄວາມຈຳຫຼັກຫຼາຍ ຊຶ່ງປະກອບດ້ວຍ ເວລາ 3 ສ່ວນ
    - ◆ ເວລາເຄື່ອນຍ້າຍຫົວອ່ານ (Mechanical movement of read/write head)
    - ◆ ເວລາໝູນເອົາຂໍ້ມູນມາຫາຫົວອ່ານ (Rotational latency)
    - ◆ ເວລາສົ່ງຂໍ້ມູນມາຫາ ໄປຫາ CPU (Transfer time)
  - ບັນດາໜ່ວຍຄວາມຈຳສຳຮອງທີ່ສາມາດຖອດອອກໄດ້ຈະຖືກໃຊ້ສຳຫຼັບ ເກັບສຳຮອງ ແລະ ເຄື່ອນຍ້າຍຂໍ້ມູນ
    - ◆ CDs (CD-R, CD-RW), DVDs (DVD-R, DVD+R), Zip disks, Floppy disks, Flash memory cards, Tapes

# ອົງປະກອບຂອງ Hardware

## ◆ ເສັ້ນທາງຂົນສົ່ງ(Buses)

- ເປັນການລວບລວມບັນດາເສັ້ນທາງສົ່ງຂໍ້ມູນຊຶ່ງເປັນສາຍໄຟນ້ອຍໆທີ່ເຊື່ອມຕໍ່ລະຫວ່າງອຸປະກອນຕ່າງໆ
- ເສັ້ນທາງຂົນສົ່ງມີດັ່ງນີ້
  - ◆ Data Bus - ສໍາຫຼັບຂົນສົ່ງຂໍ້ມູນ
  - ◆ Address Bus - ສໍາຫຼັບສົ່ງທີ່ຢູ່
  - ◆ Port - ເປັນເສັ້ນທາງຂົນສົ່ງທີ່ໃຊ້ເຊື່ອມຕໍ່ລະຫວ່າງ 2 ອຸປະກອນເທົ່ານັ້ນ
  - ◆ I/O channel - ເປັນເສັ້ນທາງຂົນສົ່ງທີ່ໃຊ້ຮ່ວມກັນຂອງບັນດາອຸປະກອນປ້ອນແລະ ສະແດງຜິນ
  - ◆ Front side bus - ເປັນເສັ້ນທາງເຊື່ອມຕໍ່ລະຫວ່າງ CPU ແລະ ໜ່ວຍຄວາມຈໍາຫຼັກ ຊຶ່ງມີຄວາມໄວ 133MHz, 200MHz
  - ◆ PCI bus - ເປັນເສັ້ນທາງເຊື່ອມຕໍ່ລະຫວ່າງ FSB ໄປຫາອຸປະກອນຕ່າງໆ
  - ◆ AGP bus - ໃຊ້ເຊື່ອມຕໍ່ກັບ Card ຈໍ



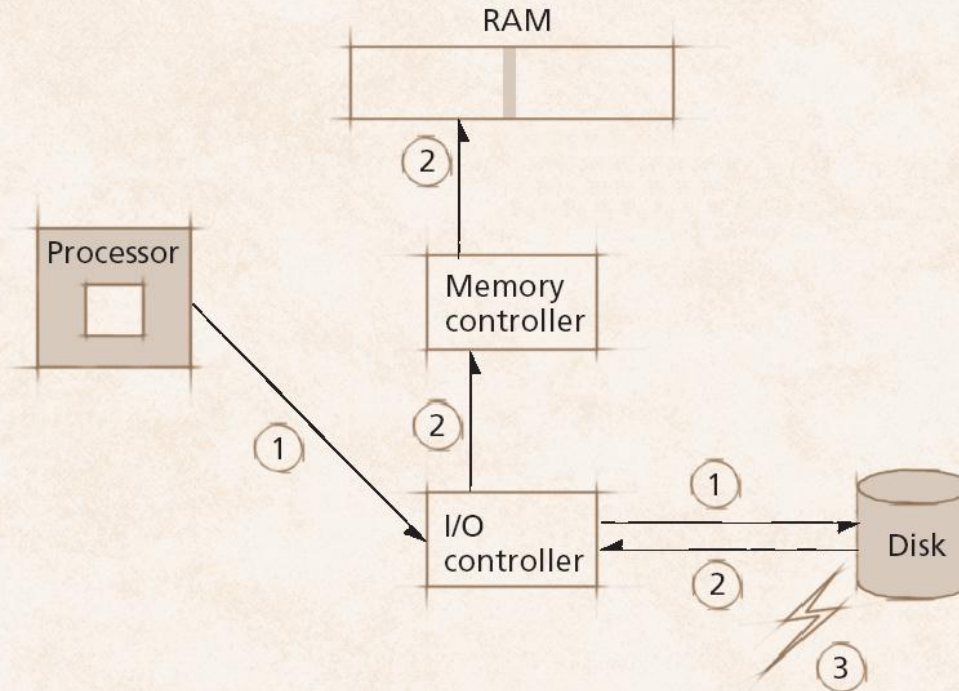
# ອົງປະກອບຂອງ Hardware

- ◆ ການເຂົ້າຫາໜ່ວຍຄວາມຈໍາໂດຍກົງ(Direct Memory Access : DMA)
  - ເປັນເທັກນິກໃນການສົ່ງຂໍ້ມູນລະຫວ່າງໜ່ວຍຄວາມຈໍາຫຼັກແລະອຸປະກອນປ້ອນ/ສະຜົນໃຫ້ໄວຂຶ້ນ
  - ໂດຍການໃຫ້ບັນດາອຸປະກອນ ແລະ ຕົວຄວບຄຸມສົ່ງຂໍ້ມູນເທື່ອລະກ້ອນໄປຫາ ແລະ ມາຈາກໜ່ວຍຄວາມຈໍາໂດຍກົງ ຊຶ່ງເປັນການເຮັດໃຫ້ CPU ມີເວລາຫວ່າງໄປເຮັດວຽກອື່ນ
  - DMA channel ຈະໃຊ້ I/O Controller ເພື່ອບໍລິຫານຈັດການການສົ່ງຂໍ້ມູນ
  - ເຮັດໃຫ້ປະສິດທິພາບຂອງລະບົບເພີ່ມຂຶ້ນໃນລະບົບທີ່ເຮັດວຽກ I/O ເປັນຈໍານວນຫຼວງຫຼາຍ

# ອົງປະກອບຂອງ Hardware



ກ  
A



- 1 A processor sends an I/O request to the I/O controller, which sends the request to the disk. The processor continues executing instructions.
- 2 The disk sends data to the I/O controller; the data is placed at the memory address specified by the DMA command.
- 3 The disk sends an interrupt to the processor to indicate that the I/O is done.



# ອົງປະກອບຂອງ Hardware

## ◆ ອຸປະກອນຕໍ່ພ່ວງທາງນອກ(Peripheral Devices)

- ແມ່ນອຸປະກອນທີ່ບໍ່ຕ້ອງການໃຫ້ຄອມພິວເຕີປະຕິບັດຄໍາສັ່ງຂອງຊອບແວຣ໌ເຊັ່ນ:
  - ◆ Network interface cards, modems, sound cards
  - ◆ Hard disk, CD and DVD drives
  - ◆ Keyboards and mice
- ບັນດາອຸປະກອນເຫຼົ່ານີ້ສາມາດຕໍ່ໃສ່ຄອມພິວເຕີໂດຍຜ່ານ Port
  - ◆ Serial ports,
  - ◆ parallel ports,
  - ◆ USB,
  - ◆ IEEE 1394 ports and
  - ◆ SCSI

# Hardware ທີ່ສະໜັບສະໜູນລະບົບປະຕິບັດການ

- ◆ ສະຖາປັດຕະຍະກຳຄອມພິວເຕີປະກອບດ້ວຍ
  - ຄຸນລັກສະນະທີ່ເຮັດໃຫ້ປະຕິບັດໜ້າທີ່ຂອງລະບົບປະຕິບັດການໄດ້ຢ່າງວ່ອງໄວໃນຮາດແວຣ໌ເພື່ອເຮັດໃຫ້ສິດທິພາບສູງຂຶ້ນ
  - ຄຸນລັກສະນະທີ່ເຮັດໃຫ້ລະບົບປະຕິບັດການມີລະບົບຄວາມປອດໄພທີ່ເຂັ້ມງວດ

# Hardware ທີ່ສະໜັບສະໜູນລະບົບປະຕິບັດການ

## ◆ ໜ່ວຍປະມວນຜົນ (Processor)

- ການບໍລິຫານ ແລະ ປ້ອງກັນໜ່ວຍຄວາມຈໍາ
  - ◆ ປ້ອງກັນຂະບວນການບໍ່ໃຫ້ເຂົ້າໄປໃຊ້ໜ່ວຍຄວາມຈໍາບ່ອນທີ່ບໍ່ໄດ້ມອບໝາຍໃຫ້ເຊັ່ນ: ໜ່ວຍຄວາມຈໍາຂອງຜູ້ອື່ນຫຼືຂອງລະບົບ
  - ◆ ເຮັດໄດ້ໂດຍການໃຊ້ register ຂອງໜ່ວຍປະມວນຜົນທີ່ສາມາດປ່ຽນແປງແກ້ໄຂໄດ້ສະເພາະຜູ້ທີ່ມີສິດ
- Interrupts ແລະ Exceptions
  - ◆ ອຸປະກອນທຸກຢ່າງຈະສົ່ງສັນຍານ Interrupt ໄປຫາໜ່ວຍປະມວນຜົນເມື່ອມີເຫດການໃດໜຶ່ງເກີດຂຶ້ນ
  - ◆ Exceptions ເປັນສັນຍານ Interrupt ທີ່ຖືກສ້າງຂຶ້ນເພື່ອຕອບສະໜອງຕໍ່ຂໍ້ຜິດພາດ

# Hardware ທີ່ສະໜັບສະໜູນລະບົບປະຕິບັດການ

## ◆ Timers ແລະ Clocks

### ■ Timers (ວົງຈອນເວລາ)

- ◆ ວົງຈອນເວລາຈະສ້າງສັນຍານ interrupt ເປັນໄລຍະ
- ◆ ລະບົບປະຕິບັດການໃຊ້ສັນຍານຂອງວົງຈອນເວລາເພື່ອປ້ອງກັນຂະບວນການບໍ່ໃຫ້ໃຊ້ໜ່ວຍປະມວນຜົນພຽງຜູ້ດຽວ

### ■ Clocks (ໂມງ)

- ◆ ເປັນສັນຍານວັດແທກຄວາມຕໍ່ເນື່ອງ
- ◆ ບອກວັນທີແລະເວລາໃນແຕ່ລະວັນ

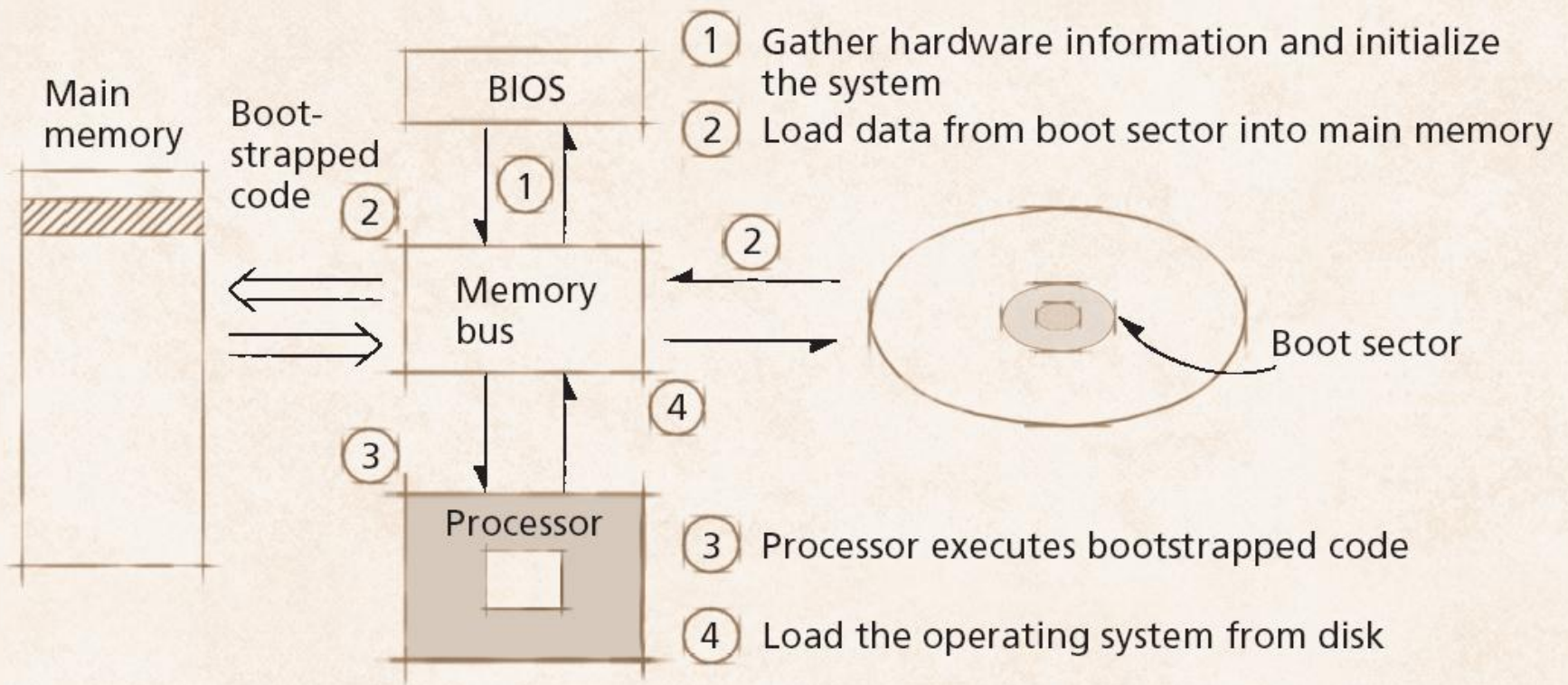
# Hardware ທີ່ສະໜັບສະໜູນລະບົບປະຕິບັດການ

## ◆ Bootstrapping

- ແມ່ນເລີ່ມຕົ້ນໂລດບັນດາພາກສ່ວນທີ່ຈຳເປັນຂອງລະບົບປະຕິບັດການເຂົ້າໄປໄວ້ໃນໜ່ວຍຄວາມຈຳ
  - ◆ ຊຶ່ງປະຕິບັດໂດຍBIOS
    - ກວກສອບຮາດແວຣ໌
    - ໂລດບັນດາຄໍາສັ່ງຕ່າງໆຈາກ boot sector ຂອງໜ່ວຍຄວາມຈຳສໍາຮອງເຂົ້າໄປໄວ້ໃນໜ່ວຍຄວາມຈຳຫຼັກ
  - ◆ ຖ້າລະບົບບໍ່ສາມາດໂລດໄດ້ຜູ້ໃຊ້ຈະບໍ່ສາມາດເອີ້ນໃຊ້ຄອມພິວເຕີໄດ້

# Hardware ທີ່ສະໜັບສະໜູນລະບົບປະຕິບັດການ

## ◆ Bootstrapping





# Hardware ທີ່ສະໜັບສະໜູນລະບົບປະຕິບັດການ

## ◆ Plug and Play

- ເປັນລະບົບທີ່ອະນຸຍາດໃຫ້ລະບົບປະຕິບັດການກຳນົດຄ່າໃຫ້ແກ່ຮາດແວຣ໌ທີ່ຕິດຕັ້ງເຂົ້າກັບເຄື່ອງຄອມພິວເຕີໃໝ່ໂດຍທີ່ຜູ້ໃຊ້ບໍ່ໄດ້ເຮັດຫຍັງ
- ເພື່ອໃຫ້ສະໜັບສະໜູນ plug and play, ຮາດແວຣ໌ຈະຕ້ອງ:
  - ◆ ສະແດງຕົວເອງຕໍ່ລະບົບປະຕິບັດການໂດຍບໍ່ຊ້ຳກັບໃຜ
  - ◆ ບອກຊັບພະຍາກອນ ແລະ ອຸປະກອນທີ່ຕ້ອງການໃນການເຮັດວຽກຂອງຕົນຕໍ່ກັບລະບົບປະຕິບັດການ
  - ◆ ສະແດງ driver ທີ່ສະໜັບສະໜູນອຸປະກອນ ແລະ ອະນຸຍາດໃຫ້ຊອບແວຣ໌ກຳນົດຄ່າໃຫ້ແກ່ອຸປະກອນນັ້ນ

# Caching ແລະ Buffering

## ◆ Caches

- ເປັນໜ່ວຍຄວາມຈໍາທີ່ມີຄວາມໄວສູງກ່ວາໝູ່ໃນບັນດາໜ່ວຍຄວາມຈໍາ
- ເກັບສໍານຶກຂໍ້ມູນທີ່ໄດ້ຖືກໃຊ້ເປັນປະຈໍາໄວ້
- ເພີ່ມຄວາມໄວໃນການເຮັດວຽກໃຫ້ແກ່ໂປຣແກຣມຕ່າງໆ
- ປະກອບດ້ວຍ:
  - ◆ L1 ແລະ L2 caches
  - ◆ ໜ່ວຍຄວາມຈໍາຫຼັກ (RAM) ອາດຈະເບິ່ງເປັນ cache ສໍາຫຼັບ hard disks ແລະ ໜ່ວຍຄວາມຈໍາສໍາຮອງອື່ນໆ



# Caching ແລະ Buffering

## ◆ Buffers

- ເປັນໜ່ວຍຄວາມຈຳຊົ່ວຄາວທີ່ເກັບຂໍ້ມູນໄວ້ໃນຂະນະທີ່ I/O ກຳລັງຂຶ້ນສິ່ງ
- ໂດຍສ່ວນໃຫຍ່ຈະໃຊ້ເພື່ອ:
  - ◆ ປະສານການເຮັດວຽກຮ່ວມກັນລະຫວ່າງບັນດາອຸປະກອນທີ່ມີຄວາມໄວຕ່າງກັນ
  - ◆ ເກັບຂໍ້ມູນໄວ້ຊົ່ວຄາວສຳຫຼັບການປະມວນຜົນທີ່ບໍ່ພ້ອມກັນ
  - ◆ ເພື່ອສົ່ງສັນຍານແບບບໍ່ພ້ອມກັນ

# Caching ແລະ Buffering

## ◆ Spooling

- ເປັນເທັກນິກການເກັບຂໍ້ມູນຊົ່ວຄາວໃຫ້ແກ່ການສື່ສານລະຫວ່າງ disk ແລະ ອຸປະກອນ I/O ທີ່ມີຄວາມໄວຕໍ່າ
- ເຮັດໃຫ້ບັນດາຂະບວນການຂໍຮ້ອງການປະຕິບັດງານກັບບັນດາອຸປະກອນຕໍ່ພ່ວງກັບຄອມພິວເຕີໂດຍທີ່ບັນດາອຸປະກອນເຫຼົ່ານັ້ນຍັງຄາວຽກອື່ນຢູ່

# ນຳສະເໜີກ່ຽວກັບ Software

## ◆ ພາສາໂປຣແກຣມ (Programming languages)

- ບາງພາສາແມ່ນຄອມພິວເຕີສາມາດເຂົ້າໃຈໄດ້ໂດຍກົງ, ບາງພາສາຈະຕ້ອງໄດ້ຜ່ານການແປກ່ອນ
- ໂດຍທົ່ວໄປຈະແບ່ງປະເພດໄດ້ດັ່ງນີ້:
  - ◆ ພາສາເຄື່ອງຄອມພິວເຕີ (Machine language)
  - ◆ ພາສາແອັສເຊມບລີ (Assembly language)
  - ◆ ພາສາລະດັບສູງ (High-level language)

# ນຳສະເໜີກ່ຽວກັບ Software



## ◆ ພາສາເຄື່ອງ ແລະ ພາສາແອັສແຊມບລີ

### ■ ພາສາເຄື່ອງຄອມພິວເຕີ

- ◆ ຂຶ້ນກັບໂດຍການອອກແບບຮາດແວຣ໌ຄອມພິວເຕີ
- ◆ ປະກອບດ້ວຍສາຍຂອງເລກຖານສອງ (1s ແລ 0s) ທີ່ບອກໃຫ້ຄອມພິວເຕີຮູ້ຈັກວິທີໃນການປະຕິບັດແບບພື້ນຖານ
- ◆ ເປັນພາສາທີ່ຄອມພິວເຕີສາມາດເຂົ້າໃຈໄດ້ໂດຍກົງ

# ນຳສະເໜີກ່ຽວກັບ Software

## ◆ ພາສາເຄື່ອງ ແລະ ພາສາແອັສເຊມບລີ

### ■ ພາສາແອັສເຊມບລີ

- ◆ ເປັນພາສາທີ່ແທນຄຳສັ່ງພາສາເຄື່ອງດ້ວຍຕົວຫຍໍ້ຂອງພາສາອັງກິດ (English-like)
- ◆ ໃຊ້ Assemblers ແປພາສາແອັສເຊມບລີໄປເປັນພາສາເຄື່ອງຄອມພິວເຕີ
- ◆ ເປັນພາສາທີ່ເຮັດວຽກໄດ້ໄວ ແລະ ບໍ່ມີຂໍ້ຜິດພາດຫຼາຍ

# ນຳສະເໜີກ່ຽວກັບ Software

## ◆ Interpreters ແລະ Compilers

- ພາສາລະດັບສູງເປັນຄຳສັ່ງທີ່ຄ້າຍຄືພາສາອັງກິດທີ່ມະນຸດໃຊ້, ສາມາດສັ່ງໃຫ້ຄອມພິວເຕີເຮັດວຽກໃດໜຶ່ງໂດຍໃຊ້ມີຈຳນວນປະໂຫຍກຄຳສັ່ງທີ່ບໍ່ຫຼາຍ ແຕ່ຕ້ອງໄດ້ແປໃຫ້ເປັນພາສາເຄື່ອງຄອມພິວເຕີ
- Compiler
  - ◆ ເປັນໂປຣແກຣມແປພາສາທີ່ໃຊ້ແປພາສາລະດັບສູງໄປເປັນພາສາເຄື່ອງ ໂດຍການແປຈະແປທັງໝົດໂປຣແກຣມເທື່ອດຽວ
- Interpreter
  - ◆ ເປັນໂປຣແກຣມແປພາສາລະດັບສູງໄປເປັນພາສາເຄື່ອງ ໂດຍການແປຈະແປເທື່ອລະຄຳສັ່ງ

# ນຳສະເໜີກ່ຽວກັບ Software

- ◆ ພາສາລະດັບສູງທີ່ໄດ້ຮັບຄວາມນິຍົມໃນໄລຍະຜ່ານມາ ແລະ ປະຈຸບັນ
  - ໂດຍປົກກະຕິເປັນພາສາແບບ procedural ຫຼື object-oriented
  - Fortran - ໃຊ້ສຳຫຼັບຂຽນໂປຣແກຣມທາງວິທະຍາສາດ ແລະ ວິສະວະກຳ
  - COBOL - ໃຊ້ສຳຫຼັບຂຽນໂປຣແກຣມທາງທຸລະກິດທີ່ຈັດການກັບຂໍ້ມູນເປັນຈຳນວນຫຼວງຫຼາຍ
  - C - ເປັນພາສາທີ່ໃຊ້ພັດທະນາລະບົບປະຕິບັດການ Unix
  - C++/Java - ເປັນພາສາແບບ object-oriented
  - C# - ເປັນພາສາໂປຣແກຣມແບບ Object-oriented ສຳຫຼັບ .NET platform

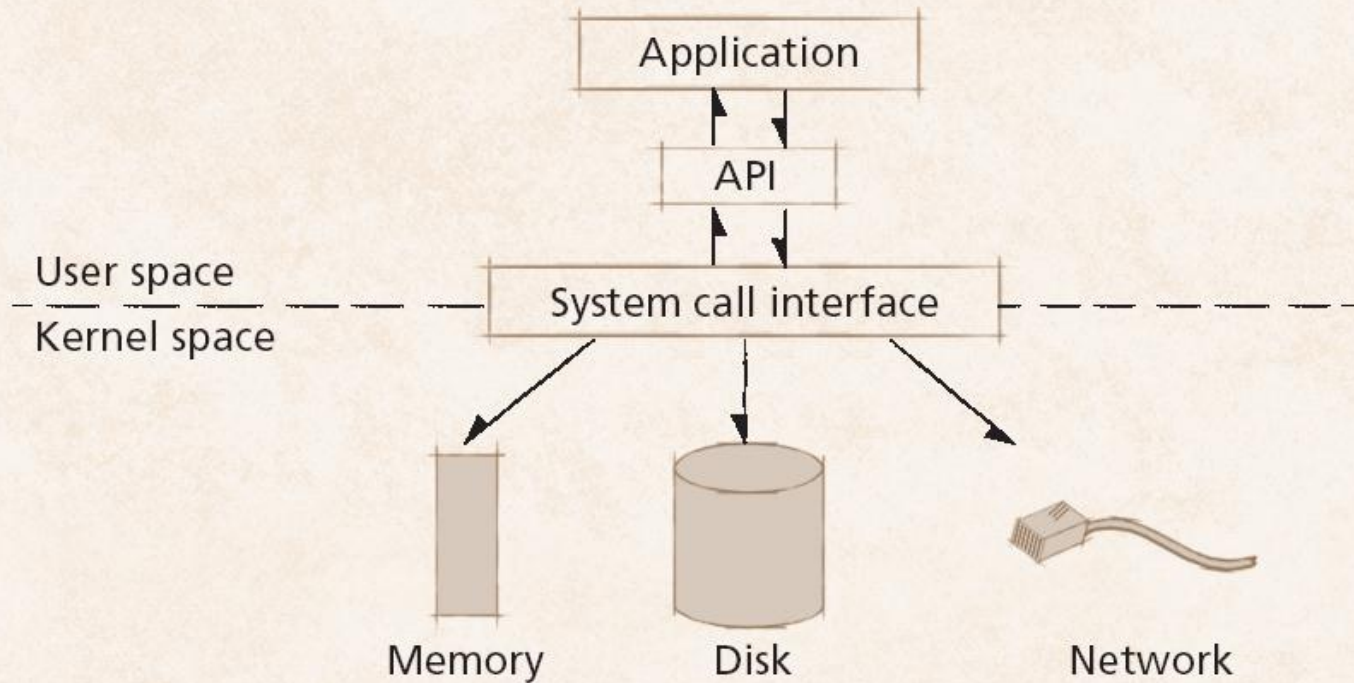
# Application Programming Interface (API)

## ◆ ເປັນຊຸດຂອງຄໍາສັ່ງ

- ນັກຂຽນໂປຣແກຣມໃຊ້ບັນດາຄໍາສັ່ງເຫຼົ່ານີ້ເພື່ອຂໍບໍລິການຈາກລະບົບປະຕິບັດການ
- ໂປຣແກຣມ API ສາມາດເອີ້ນໃຊ້ລະບົບປະຕິບັດການໂດຍໃຊ້ system calls
- ຕົວຢ່າງຂອງ APIs ແມ່ນ:
  - ◆ Portable Operating System Interface (POSIX)
  - ◆ Windows API



# Application Programming Interface (API)



# Compiling, Linking ແລະ Loading

- ◆ ກ່ອນພາສາລະດັບສູງຈະສາມາດເຂົ້າໃຈໄດ້ດ້ວຍຄອມພິວເຕີ ມັນຈະຕ້ອງ:
  - ຖືກແປໄປເປັນພາສາເຄື່ອງຄອມພິວເຕີ
  - ເຊື່ອມຍິງ (Link) ມັນເຂົ້າກັບພາສາເຄື່ອງສ່ວນອື່ນໆທີ່ມັນຂຶ້ນກັບ
  - ໂລດມັນເຂົ້າມາເກັບໄວ້ໃນໜ່ວຍຄວາມຈຳຫຼັກ

# Firmware

- ◆ Firmware ບັນຈຸຄໍາສັ່ງທີ່ສາມາດເຮັດວຽກໄດ້ (executable) ທີ່ຖືກເກັບໄວ້ໃນໜ່ວຍຄວາມຈໍາທີ່ບໍ່ປ່ຽນແປງຢູ່ໃນອຸປະກອນໃດໜຶ່ງ
  - Programmed with microprogramming
    - ◆ Layer of programming below a computer's machine-language
    - ◆ Microcode
      - Simple, fundamental instruction necessary to implement all machine-language operations

# Middleware

## ◆ ເປັນຊອບແວສໍາຫຼັບ distributed systems

- ເປັນຕົວກາງໃນການເຮັດວຽກລະຫວ່າງບັນດາຊອບແວຕ່າງໆທີ່ເຮັດວຽກຢູ່ໃຈເຄື່ອງຄອມພິວເຕີເຄື່ອງໜຶ່ງຫຼື ຫຼາຍເຄື່ອງຜ່ານລະບົບເຄືອຂ່າຍ
- ອໍານວຍຄວາມສະດວກໃຫ້ແກ່ບັນດາລະບົບກະຈາຍທີ່ແຕກຕ່າງກັນ
- ເຮັດໃຫ້ງ່າຍຕໍ່ການເຮັດວຽກຂອງການຂຽນໂປຣແກຣມເຮັດວຽກຜ່ານລະບົບເຄືອຂ່າຍ
- ຕົວຢ່າງ: ການເປີດການເຊື່ອມຕໍ່ກັບຖານຂໍ້ມູນ (Open DataBase Connectivity: ODBC)
  - ◆ ອະນຸຍາດໃຫ້ໂປຣແກຣມເຂົ້າໄປຫາຖານຂໍ້ມູນໂດຍຜ່ານ middleware ທີ່ຊື່ວ່າ ODBC driver