

ລະບົບປະຕິບັດການ (Operating System:OS)

ອາຈານສອນ: ບົວສິດ ໄຊຍະຈັກ

ອຸດທິການສຶກສາ: ປະລິຍາໂທວິທະຍາສາດ
(ວິທະຍາສາດຄອມພິວເຕີ)

ໂທລະສັບມືຖື: (+856-20) 22245134

ອີເມລ: bouasoth@yahoo.com

ບົດທີ 1

ຄວາມຮູ້ເບື້ອງຕົ້ນກ່ຽວກັບລະບົບ
ປະຕິບັດການ

ເນື້ອໃນຫຍໍ້

- ◆ ສະເໜີເບື້ອງຕົ້ນ
- ◆ ນິຍາມຂອງລະບົບປະຕິບັດການ
- ◆ ປະຫວັດຄວາມເປັນມາຂອງລະບົບປະຕິບັດການ
- ◆ ປະຫວັດຄວາມເປັນມາຂອງ Internet ແລະ World Wide Web
- ◆ ເປົ້າໝາຍ ແລະ ອົງປະກອບຂອງລະບົບປະຕິບັດການ
- ◆ ສະຖາປັດຕະຍະກຳຂອງລະບົບປະຕິບັດການ

ສະເໜີເບື້ອງຕົ້ນ

- ◆ ຄອມພິວເຕີໄດ້ຂະຫຍາຍຕົວຢ່າງໄວວາໃນໄລຍະຫຼາຍທົດສະວັດຜ່ານມາ
- ◆ ປະສິດທິພາບຂອງຄອມພິວເຕີໄດ້ເພີ່ມຂຶ້ນຢ່າງໄວວາ ໃນຂະນະທີ່ລາຄາຂອງມັນລຸດລົງກະທັນຫັນ
- ◆ ໃນປະຈຸບັນເຄື່ອງ PC ໄດ້ມີຄວາມສາມາດເຮັດວຽກໄດ້ຫຼາຍພັນລ້ານຄໍາສັ່ງໃນໜຶ່ງວິນາທີ
- ◆ Super Computer ສາມາດເຮັດວຽກໄດ້ຫຼາຍລ້ານລ້ານຄໍາສັ່ງໃນໜຶ່ງວິນາທີ
- ◆ ໜ່ວຍປະມວນຜົນມີປະສິດທິພາບສູງຂຶ້ນ ແລະ ລາຄາຖືກລົງ
- ◆ ຄອມພິວເຕີໄດ້ເຂົ້າມາມີບົດບາດໃນທຸກສາຂາວຽກງານ ແລະ ໃນຊີວິດປະຈຳວັນຂອງມະນຸດ

ນິຍາມຂອງລະບົບປະຕິບັດການ

- ❑ ລະບົບປະຕິບັດການແມ່ນຊຸດຄໍາສັ່ງທີ່ໃຊ້ຄວບຄຸມຮາດແວຣ໌ຄອມພິວເຕີ ແລະ ການເຮັດວຽກທັງໝົດໃນລະບົບ
- ❑ ເປັນກຸ່ມຂອງໂປຣແກຣມທີ່ເຮັດໜ້າທີ່ເປັນສື່ກາງລະຫວ່າງຜູ້ໃຊ້ກັບເຄື່ອງຄອມພິວເຕີເພື່ອໃຫ້ການໃຊ້ຄອມພິວເຕີມີຄວາມສະດວກ ແລະ ເຮັດໃຫ້ການໃຊ້ Hardware ມີປະສິດທິພາບສູງສຸດ
- ❑ ເປັນສື່ກາງລະຫວ່າງຊອບແວຣ໌ນໍາໃຊ້ກັບຮາດແວຣ໌
- ❑ ເປັນຜູ້ບໍລິຫານຈັດການຊັບພະຍາກອນທັງໝົດໃນລະບົບທັງຮາດແວຣ໌ ແລະ ຊອບແວຣ໌

ປະຫວັດຄວາມເປັນມາຂອງລະບົບປະຕິບັດການ

◆ ໃນຊ່ວງປີ 1940s ແລະ 1950s

- ລະບົບປະຕິບັດການໄດ້ຖືກພັດທະນາເປັນຫຼາຍຂັ້ນຕອນ
- ໃນຊ່ວງປີ 1940, ຄອມພິວເຕີບໍ່ທັນມີລະບົບປະຕິບັດການ
- ໃນຊ່ວງປີ 1950, ລະບົບປະຕິບັດການສາມາດ
 - ◆ ເປີດໄດ້ເທື່ອລະໂປຣແກຣມ
 - ◆ ມີລະບົບສິ່ງຕໍ່ການຄວບຄຸມຈາກວຽກໜຶ່ງໄປຫາວຽກໜຶ່ງ
 - ◆ ເປັນລະບົບທ້ອນໂຮມວຽກເຂົ້າປະມວນຜົນເປັນກຸ່ມ
 - ◆ ໂປຣແກຣມແລະຂໍ້ມູນຖືກເກັບໄວ້ເປັນລຳດັບໃນເທບ

ປະຫວັດຄວາມເປັນມາຂອງລະບົບປະຕິບັດການ

◆ ໃນຊ່ວງປີ 1960s

- ຍັງເປັນລະບົບທ້ອນໂຮມວຽກເຂົ້າປະມວນຜົນເປັນກຸ່ມ
- ແຕ່ສາມາດເປີດໄດ້ເທື່ອລະຫຼາຍໂປຣແກຣມ
- ໃນປີ 1964, ບໍລິສັດ IBM ໄດ້ສະເຄື່ອງຄອມພິວເຕີລຸ້ນໃໝ່ທີ່ຊື່ວ່າ System/360
- ລະບົບ Time Sharing ໄດ້ຖືກພັດທະນາຂຶ້ນມາໃຫ້ສາມາດໃຫ້ບໍລິການແກ່ຫຼາຍຜູ້ໃຊ້ໃນເວລາດຽວກັນ
 - ◆ ເວລາໂດຍລວມໃນການເຮັດວຽກລຸດລົງ
- ລະບົບ Real Time ໄດ້ຖືກພັດທະນາຂຶ້ນມາເພື່ອໃຫ້ເຮັດວຽກທັນທີທີ່ສັງການ

ປະຫວັດຄວາມເປັນມາຂອງລະບົບປະຕິບັດການ

◆ ໃນຊ່ວງປີ 1970s

- ເປັນລະບົບ multimode timesharing ທີ່ສະໜັບສະໜູນທັງໂປຣແກຣມໃນແບບ batch processing, timesharing ແລະ real-time
- ເລີ່ມຕົ້ນມີລະບົບຄອມພິວເຕີຕັ້ງໂຕະ (Desktop)
- ກະຊວງການຕ່າງປະເທດສະຫະລັດໄດ້ພັດທະນາ TCP/IP ຂຶ້ນມາໃຫ້ເປັນມາດຕະຖານການສື່ສານລະຫວ່າງຄອມພິວເຕີ

ປະຫວັດຄວາມເປັນມາຂອງລະບົບປະຕິບັດການ

◆ ໃນຊ່ວງປີ 1980s

- ເປັນທິດສະຫວັດແຫ່ງ personal computers ແລະ workstations
- ວິທະຍາການຄອມພິວເຕີໄດ້ຖືກນຳໃຊ້ໃນໂຂງເຂດຕ່າງໆທີ່ຕ້ອງການມັນ
- PC ສາມາດຮຽນຮູ້ ແລະ ໃຊ້ງານງ່າຍເພາະວ່າເປັນ GUI
- ການເອົາຂໍ້ມູນຈາກເຄື່ອງໜຶ່ງໄປຫາອີກເຄື່ອງໜຶ່ງແມ່ນໃຊ້ລະບົບເຄືອຂ່າຍຊຶ່ງເຮັດໃຫ້ປະຢັດຄ່າໃຊ້ຈ່າຍ
- ລະບົບຄອມພິວເຕີແບບ Client/server ກຳລັງໄດ້ຮັບຄວາມນິຍົມຢ່າງກ້ວາງຂວາງ
- ວິສະວະກຳຊອບແວຮຸ້ນສືບຕໍ່ພັດທະນາຕາມລຳດັບ

ປະຫວັດຄວາມເປັນມາຂອງລະບົບປະຕິບັດການ

◆ ໃນຊ່ວງປີ 1990s

- ປະສິດທິພາບຂອງຮາດແວໄດ້ເພີ່ມຂຶ້ນຢ່າງກະທັນຫັນ
 - ◆ ໜ່ວຍປະມວນຜົນມີປະສິດທິພາບສູງຂຶ້ນ ແລະ ໜ່ວຍຄວາມຈໍາໃຫຍ່ຂຶ້ນ ແຕ່ລາຄາຖືກລົງຫຼາຍ
 - ◆ ໜ່ວຍປະມວນຜົນສາມາດປະຜົນໂປຣແກຣມທີ່ໃຫຍ່ ແລະ ສະຫຼັບຊັບຊ້ອນເທິງເຄື່ອງ PC
 - ◆ ເຄື່ອງຄອມພິວເຕີລາຄາຖືກລົງ, ສາມາດບັນຈຸຖານຂໍ້ມູນຂະໜາດໃຫຍ່ຂຶ້ນ ແລະ ປະມວນຜົນວຽກໄດ້ຫຼາຍຂຶ້ນ
 - ◆ Mainframes ຖືກໃຊ້ໜ້ອຍລົງ
- ຄອມພິວເຕີໄດ້ກ້າວເຂົ້າຫາຍຸກຂອງລະບົບແບບກະຈາຍ

ປະຫວັດຄວາມເປັນມາຂອງລະບົບປະຕິບັດການ

◆ ໃນຊ່ວງປີ 1990s

- ຄອມພິວເຕີທັງໝົດສະໜັບສະໜູນການເຮັກຜ່ານເຄືອຂ່າຍ
- ບໍລິສັດໄມໂຄຣຊອບໄດ້ຄອບຄອງຕະຫຼາຍເປັນສ່ວນໃຫຍ່
 - ◆ ລະບົບປະຕິບັດການວິນໂດມິຄູນລັກສະນະທຸກຢ່າງທີ່ມີໃນລະບົບ Macintosh ແຕ່ກ່ອນ
 - ◆ ຜູ້ໃຊ້ສາມາດໃຊ້ຫຼາຍໂປຣແກຣມໃນເວລາດຽວກັນ
- Object technology ເປັນທີ່ນິຍົມຫຼາຍຂຶ້ນ
 - ◆ ຫຼາຍໂປຣແກຣມທີ່ຂຽນຈາກພາໂປຣແກຣມແບບ Object
 - ◆ Object-oriented operating systems (OOOS) ໄດ້ຖືກພັດທະນາຂຶ້ນມາ

ປະຫວັດຄວາມເປັນມາຂອງລະບົບປະຕິບັດການ

◆ ໃນຊ່ວງປີ 1990s

- ຊອບແວທີ່ມີລາຍເຊັນເກືອບທັງໝົດຂ້າຍເປັນ object code
 - ◆ ບໍ່ລວມຊອດໂຄດ
 - ◆ ຜູ້ຜະລິດຈະເຊື່ອງຊ້ອນລາຍລະອຽດ ແລະ ເທັກນິກ
- ຟຣີແລະໂອເຟິນຊອດກຳລັງໄດ້ຮັບການພັດທະນາຫຼາຍຂຶ້ນ
 - ◆ ມີຊອດໂຄດໃຫ້ພ້ອມ
- Richard Stallman ເປີດໂຄງການ GNU
 - ◆ ສ້າງ ແລະ ຂະຫຍາຍເຄື່ອງມືຂອງລະບົບ UNIX ຄືນໃໝ່

ປະຫວັດຄວາມເປັນມາຂອງລະບົບປະຕິບັດການ

◆ ໃນຊ່ວງປີ 1990s

- Open Source ກຳລັງເລີ່ມຕົ້ນພັດທະນາ
 - ◆ ເຫັນວ່າຈະໄດ້ຮັບປະໂຫຍດໃນອະນາຄົດ
 - ◆ ເປັນການອຳນວຍຄວາມສະດວກໃຫ້ແກ່ການພັດທະນາຊອບແວຣ໌ ເພາະວ່າທຸກຄົນຈະເປັນຜູ້ທົດສອບ, ແກ້ໄຂຂໍ້ມົດພາດ, ຂະຫຍາຍ
 - ◆ ຂໍ້ບົກພ່ອງຂອງລະບົບທຸກຢ່າງຈະກວດພົບ ແລະ ຖືກແກ້ໄຂ
 - ◆ ທຸກອົງກອນສາມາດປັບປຸງແກ້ໄຂຊອດໂຄດໃຫ້ດີຂຶ້ນ
- ລະບົບປະຕິບັດການມີຄວາມເປັນມິດຕໍ່ໃຊ້ຫຼາຍຂຶ້ນ
 - ◆ ເປັນລະບົບ GUI ແລະ “Plug-and-play”

ປະຫວັດຄວາມເປັນມາຂອງລະບົບປະຕິບັດການ

◆ ໃນຊ່ວງປີ 2000 ແລະ ໃນອານາຄົດ

- ມີຊອບແວທີ່ເປັນຕົວກາງໃນການປະສານງານລະຫວ່າງລະບົບຕ່າງໆທີ່ແຕກຕ່າງກັນໃຫ້ເຮັດວຽກຮ່ວມກັນໄດ້ຜ່ານລະບົບເຄືອຂ່າຍຊຶ່ງເອີ້ນວ່າ **Middleware**
 - ◆ ຈະຖືກໃຊ້ໂຫຍດໃນລະບົບ **Web services**
- ລະບົບ **Web services** ໄດ້ພັດທະນາ
 - ◆ ເປັນການລວມເອົາບັນດາມາດຕະຖານຕ່າງໆເຂົ້າກັນ
 - ◆ ເປັນຊອບແວທີ່ກຽມພ້ອມໃຊ້ງານຜ່ານລະບົບ **Internet** ໄດ້
 - ◆ ເຮັດໃຫ້ບັນດາໂປຣແກຣມສາມາດສື່ສານ ແລະ ແລກປ່ຽນຂໍ້ມູນກັນໄດ້

ປະຫວັດຄວາມເປັນມາຂອງ Internet ແລະ World Wide Web

- ◆ Advanced Research Projects Agency (ARPA) ໄດ້ສ້າງ ARPAnet ຂຶ້ນມາໃນຊ່ວຍທ້າຍຊຸມປີ 1960s
- ◆ ARPAnet ໄດ້ກາຍເປັນປູ່ຂອງລະບົບ Internet ໃນປະຈຸບັນ
- ◆ ARPA ໄດ້ພັດທະນາ TCP/IP ເພື່ອເປັນມາດຕະຖານໃນການສື່ສານລະຫວ່າງຄອມພິວເຕີໃນ ARPAnet
- ◆ TCP/IP ຈະບໍລິຫານຈັດການການສື່ສານລະຫວ່າງໂປຣແກຣມຕ່າງໆ ໂດຍຮັບປະກັນວ່າຂໍ້ມູນຈະຖືກສົ່ງຈາກຕົ້ນທາງຫາປາຍທາງຢ່າງຖືກຕ້ອງ
- ◆ ໃນຕອນຫຼັງມາ TCP/IP ໄດ້ຖືເອົາມານຳໃຊ້ໃນວຽກງານຄ້າຂາຍ

ປະຫວັດຄວາມເປັນມາຂອງ Internet ແລະ World Wide Web

- ◆ World Wide Web (WWW) ໄດ້ຖືກພັດທະນາຂຶ້ນມາໃນຊ່ວງທ້າຍປີ 1989 ທີ່ CERN ໂດຍ Tim Berners-Lee
- ◆ ໃຊ້ເພື່ອບອກທີ່ຕັ້ງ ແລະ ເປີດເບິ່ງເອກະສານທີ່ປະກອບດ້ວຍລະບົບ multimedia-based ໃນທຸກກໍລະນີ
- ເປັນເທັກໂນໂລຢີໃນການໃຊ້ຂໍ້ມູນຂ່າວສານຮ່ວມກັນໂດຍຜ່ານເອກະສານ hyperlinked ທີ່ສ້າງຈາກ HyperText Markup Language ໂດຍຜ່ານ Hypertext Transfer Protocol (HTTP)

ເປົ້າໝາຍ ແລະ ອົງປະກອບຂອງລະບົບ ປະຕິບັດການ

- ◆ ລະບົບຄອມພິວເຕີໄດ້ຖືກພັດທະນາຈາກທີ່ບໍ່ມີລະບົບປະຕິບັດການ ຈຶ່ງກາຍເປັນລະບົບ multiprogramming ແລະ timesharing, ມີ PC ແລະ ຕອນທ້າຍລະບົບກະຈາຍໄດ້ຖືກພັດທະນາຂຶ້ນມາ
- ◆ ຜູ້ໃຊ້ສາມາດຕິດຕໍ່ປະສານງານກັບລະບົບປະຕິບັດການໂດຍໃຊ້ໂປຣແກຣມພິເສດທີ່ຊື່ວ່າ shell
- ◆ ໃນລະບົບປະຕິບັດການຈະມີກຸ່ມໂປຣແກຣມທີ່ຊື່ວ່າ Kernel ທີ່ເປັນຊອບແວທີ່ບັນຈຸບັນດາອົງປະກອບໜັກຂອງລະບົບປະຕິບັດການ

ເປົ້າໝາຍ ແລະ ອົງປະກອບຂອງລະບົບ ປະຕິບັດການ

- ◆ ໂດຍປົກກະຕິອົງປະກອບຂອງລະບົບປະຕິບັດການປະກອບດ້ວຍ
 - Processor scheduler
 - Memory manager
 - I/O manager
 - Interprocess communication (IPC) manager
 - File system manager

ເປົ້າໝາຍ ແລະ ອົງປະກອບຂອງລະບົບ ປະຕິບັດການ

- ◆ ໃນສະພາບແວດລ້ອມຂອງລະບົບ Multiprogramm ຈະມີມີລັກສະນະດັ່ງນີ້
 - Kernel ຈະເປັນຜູ້ບໍລິຫານຈັດການການເຮັດວຽກຂອງໂປຣແກຣມຕ່າງໆ
 - ໂປຣແກຣມອາດຈະປະກອບດ້ວຍພາກສ່ວນຕ່າງໆເຮັດວຽກເປັນອິດສະລະຕໍ່ກັນ, ແຕ່ໃຊ້ໜ່ວຍຄວາມຈໍາຮ່ວມກັນເພື່ອໃຊ້ຂໍ້ມູນຮ່ວມກັນຊຶ່ງເອີ້ນວ່າ threads
 - ເພື່ອໃຊ້ງານບັນດາອຸປະກອນ I/O, ໂປຣແກຣມຕ່າງໆຈະຕ້ອງເອີ້ນໃຊ້ System call ຊຶ່ງຈັດການໂດຍ driver ອຸປະກອນຕ່າງໆທີ່ບັນຈຸຄໍາສັ່ງສະເພາະແຕ່ລະອຸປະກອນເພື່ອເຂົ້າໄປໃຊ້ຮາດແວຣ໌ໂດຍກົງ

ເປົ້າໝາຍ ແລະ ອົງປະກອບຂອງລະບົບ ປະຕິບັດການ

◆ ເປົ້າໝາຍຂອງລະບົບປະຕິບັດການ

- **Efficiency** - ເຮັດວຽກໄດ້ຫຼາຍທີ່ສຸດ ແລະ ໃຊ້ເວລາໜ້ອຍ
- **Robustness** - ປ້ອງກັນຄວາມຜິດພາດ ແລະ ມີຄວາມໜ້າເຊື່ອຖື
- **Scalability** - ສາມາດເພີ່ມຊັບພະຍາກອນຕ່າງໆຕາມຕ້ອງການ
- **Extensibility** - ສາມາດເຮັດວຽກກັບເທັກໂນໂລຍີໄດ້
- **Portability** - ສາມາດເຮັດວຽກກັບຮາດແວຮ່ຕ່າງໆໄດ້
- **Security** - ປ້ອງກັນຜູ້ໃຊ້ ແລະ ຊອບແວຮ່ເຂົ້າໃຊ້ອຸປະກອນ ແລະ ຊັບພະຍາກອນໂດຍບໍ່ໄດ້ຮັບອະນຸຍາດ
- **Protection** - ເທັກນິກທີ່ໃຊ້ນະໂຍບາຍຄວາມປອດໄພ
- **Interactivity** - ເຮັດໃຫ້ໂປຣແກຣມຕອບສະໜອງຕໍ່ຜູ້ໃຊ້ໄດ້ໄວ
- **Usability** - ໃຫ້ເປັນປະໂຫຍດຕໍ່ຜູ້ໃຊ້

ສະຖາປັດຕະຍະກຳຂອງລະບົບປະຕິບັດການ

- ◆ ລະບົບປະຕິບັດການໃນປະຈຸບັນມີແນວໂນ້ມສະລັບຊັບຊ້ອນຂຶ້ນເລື້ອຍໆ
 - ໃຫ້ບໍລິການໄດ້ຫຼາຍອັນ
 - ສະໜັບສະໜູນຮາດແວຣ໌ ແລະ ຊອບແວຣ໌ທີ່ແຕກຕ່າງກັນ
 - ໂຄງສ້າງຂອງລະບົບປະຕິບັດການຈະຊ່ວຍຈັດການຄວາມຊັບຊ້ອນດັ່ງກ່າວ
 - ◆ ຈັດສັນອົງປະກອບຂອງລະບົບປະຕິບັດການໃຫ້ເປັນລະບຽບ
 - ◆ ກຳໜົດສິດທິພິເສດໃຫ້ແກ່ແຕ່ລະພາກສ່ວນໃນການເຮັດວຽກ

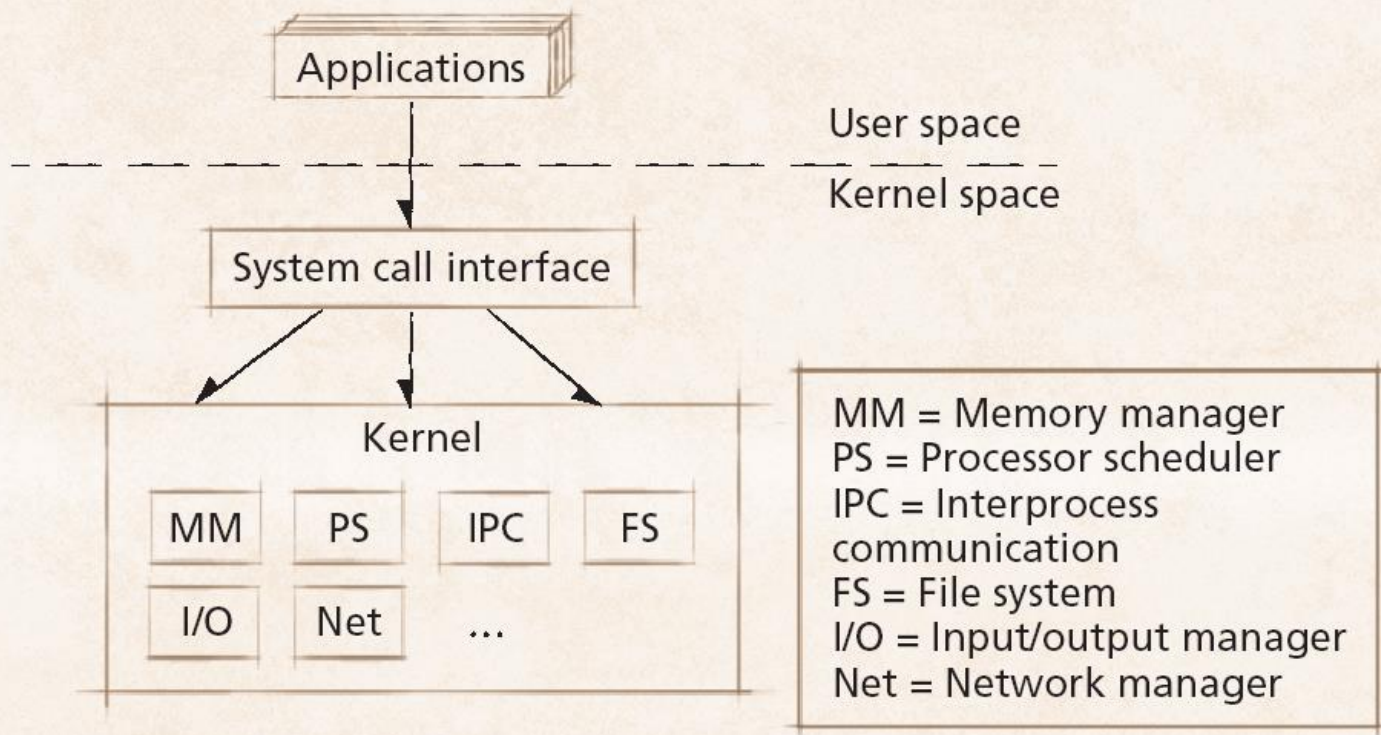
ສະຖາປັດຕະຍະກຳຂອງລະບົບປະຕິບັດການ

◆ ສະຖາປັດຕະຍະກຳແບບ Monolithic

- ທຸກອົງປະກອບຂອງລະບົບປະຕິບັດການຈະຖືກບັນຈຸໄວ້ໃນ kernel ແລະ ແຕ່ລະສ່ວນສາມາດສື່ສານກັນໄດ້ໂດຍກົງ
- ການເຮັດວຽກມີປະສິດທິພາບສູງ
- ເມື່ອເກີດມີຂໍ້ຜິດພາດເລັກໜ້ອຍຈະກວດໄດ້ຍາກ

ສະຖາປັດຕະຍະກຳຂອງລະບົບປະຕິບັດການ

◆ ສະຖາປັດຕະຍະກຳແບບ Monolithic



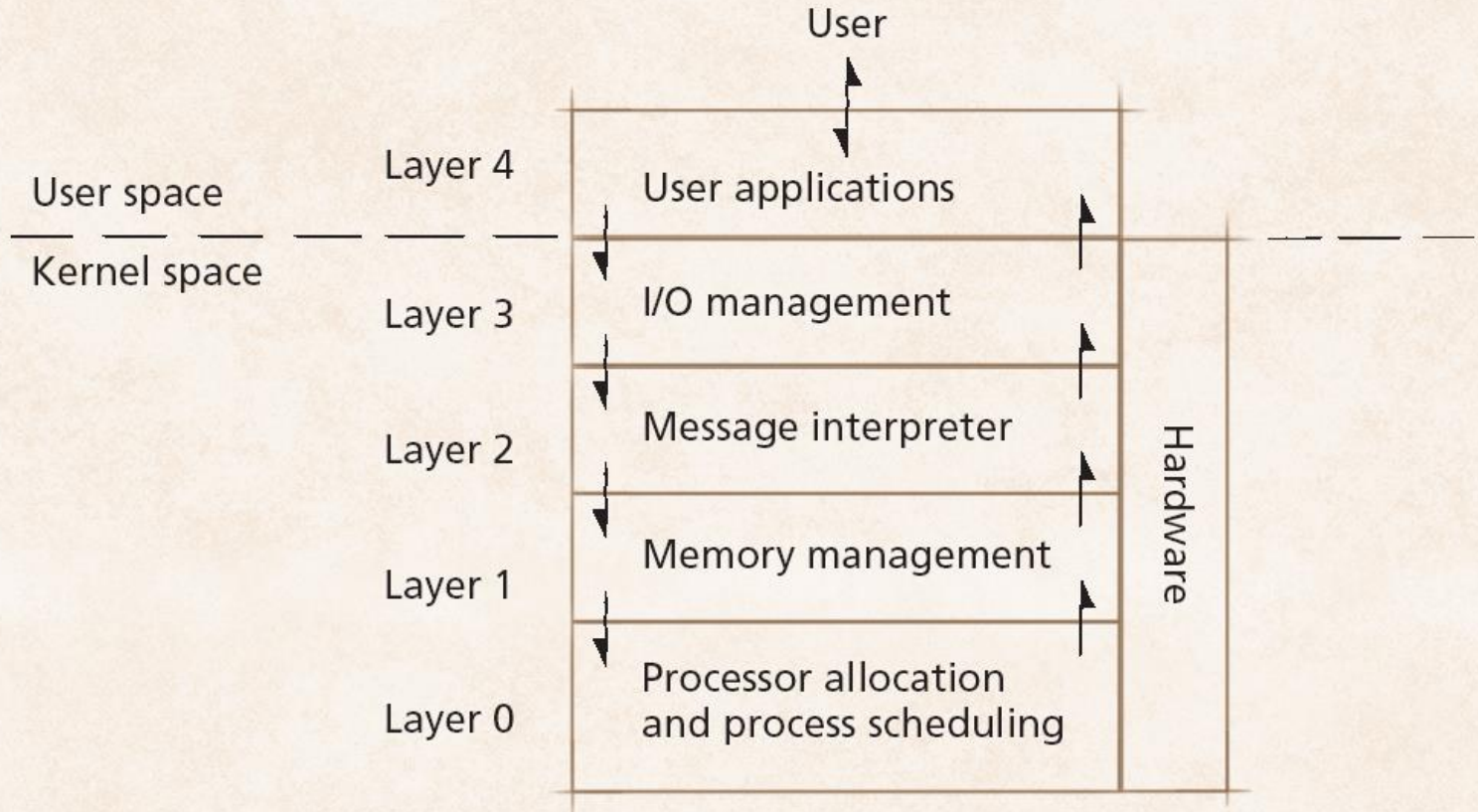
ສະຖາປັດຕະຍະກຳຂອງລະບົບປະຕິບັດການ

◆ ສະຖາປັດຕະຍະກຳແບບ Layer

- ອອກແບບມາເພື່ອປັບປຸງຜົນເສຍຂອງ Monolithic ໂດຍການຈັດກຸ່ມ ບັນດາອົງປະກອບທີ່ເຮັດວຽກຄ້າຍຄືກັນໄວ້ໃນຊັ້ນດຽວກັນ
- ແຕ່ລະຊັ້ນສາມາດສື່ສານໄດ້ໂດຍກົງກັບຊັ້ນທີ່ຢູ່ລຸ່ມແລະເທິງທີ່ຕິດກັບ ມັນເທົ່ານັ້ນ
- ການເອີ້ນໃຊ້ຂະບວນການຈະຕ້ອງຜ່ານຫຼາຍຊັ້ນກ່ອນສຳເລັດ
- ຜົນການປະມວນຜົນໄດ້ໜ້ອຍກ່ວາ Monolithic, ສະນັ້ນ ຈຶ່ງຕ້ອງໄດ້ມີ ວິທີການໃໝ່ໃນການສົ່ງຂໍ້ມູນ ແລະ ຄຳສັ່ງຄວບຄຸມ

ສະຖາປັດຕະຍະກຳຂອງລະບົບປະຕິບັດການ

◆ ສະຖາປັດຕະຍະກຳແບບ Layer

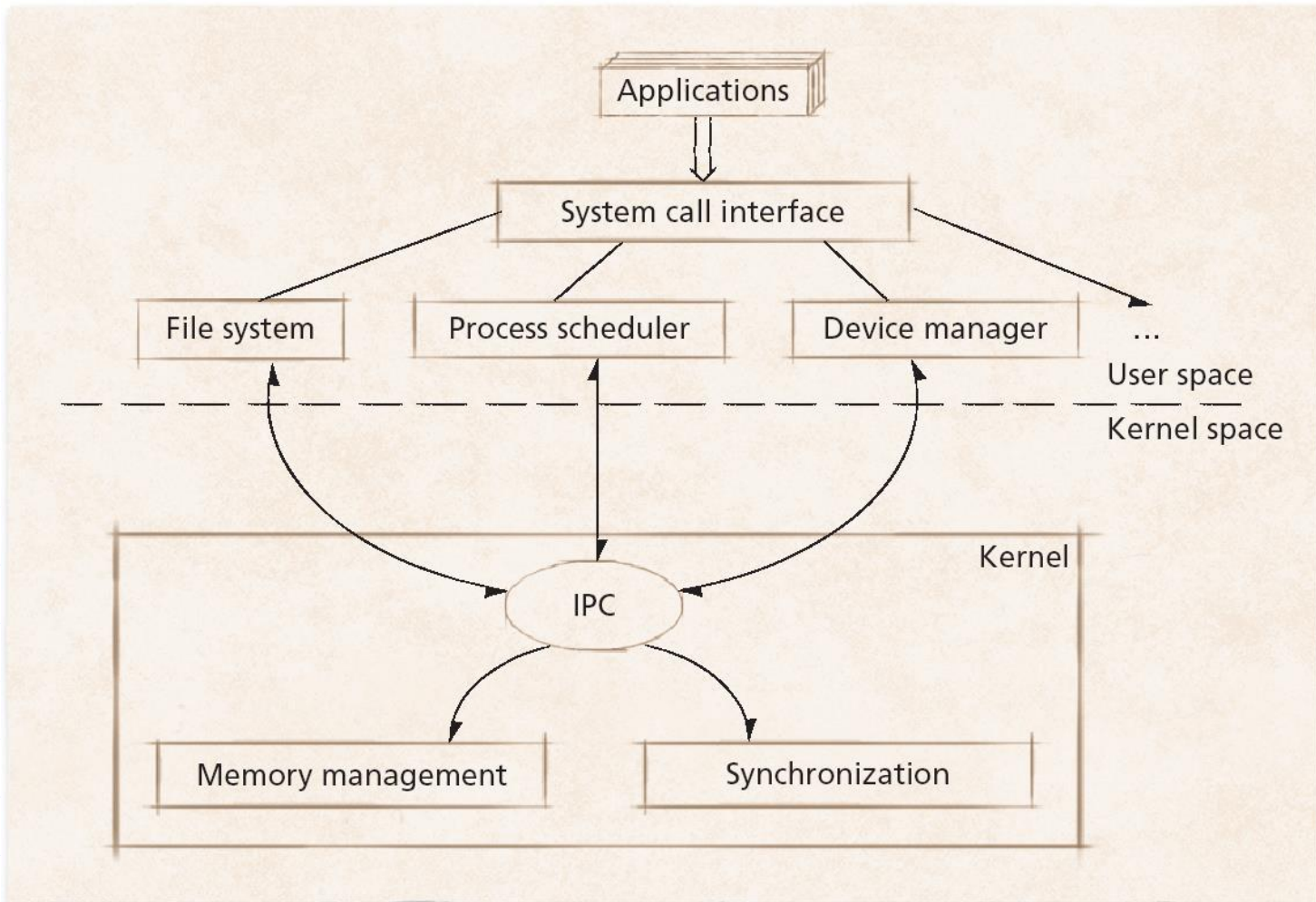


ສະຖາປັດຕະຍະກຳຂອງລະບົບປະຕິບັດການ

◆ ສະຖາປັດຕະຍະກຳແບບ Microkernel

- ສາມາດໃຫ້ບໍລິການໄດ້ຈຳນວນໜຶ່ງເທົ່ານັ້ນ ຊຶ່ງເປັນການເຮັດໃຫ້ kernel ນ້ອຍທີ່ສຸດ ແລະ ເພີ່ມຊັບພະຍາກອນໄດ້ຕາມຕ້ອງການ
- ມີລະດັບຄວາມເປັນໂມດູນສູງ (High degree of modularity)
 - ◆ ເປັນບົບທີ່ Extensible, portable ແລະ scalable
- ເຮັດໃຫ້ມີການສື່ສານລະຫວ່າງໂມດູນຫຼາຍຂຶ້ນ, ສະນັ້ນ ອາດຈະເຮັດໃຫ້ປະສິດທິພາບຂອງລະບົບລຸດລົງ

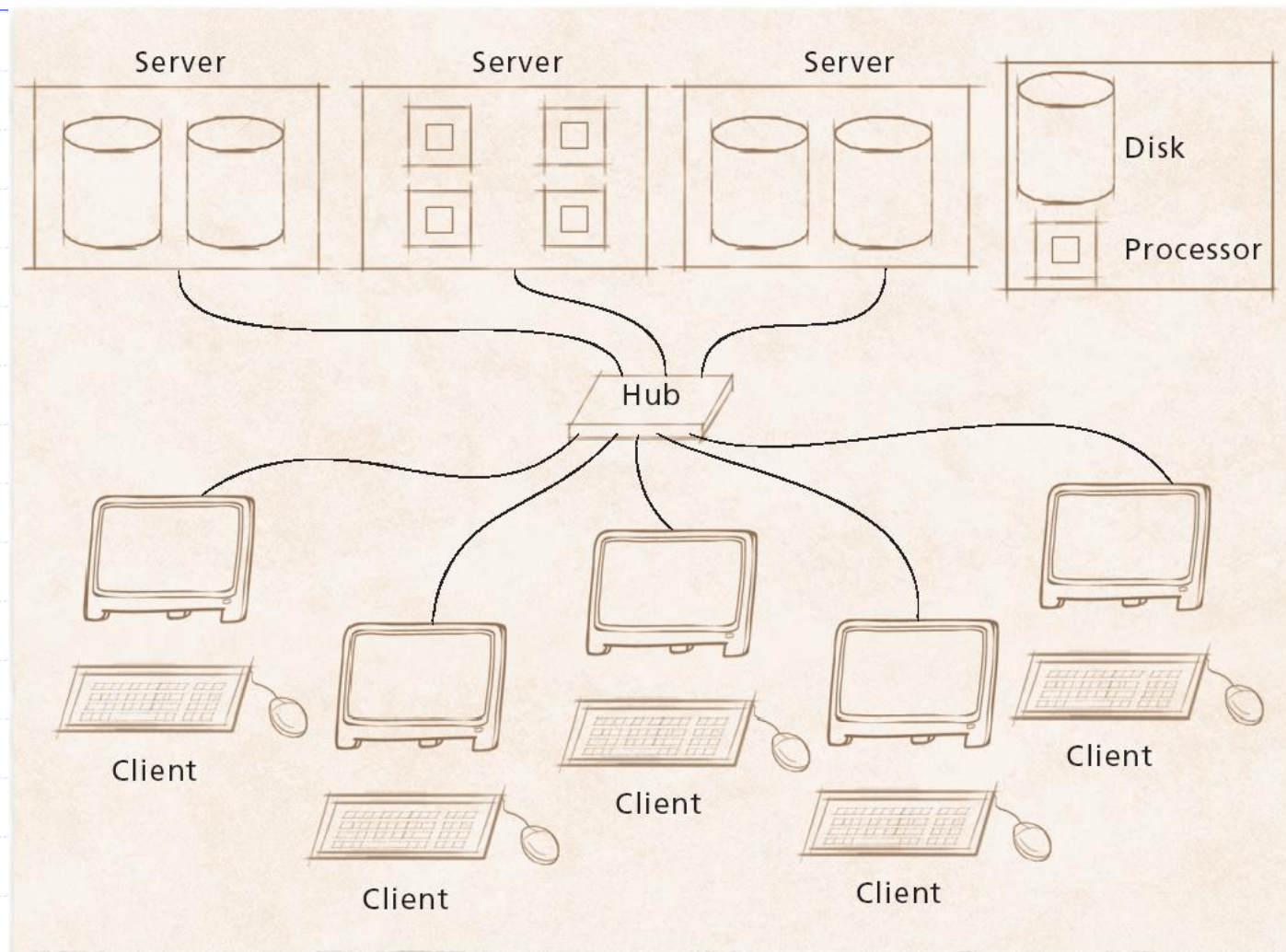
ສະຖາປັດຕະຍະກຳຂອງລະບົບປະຕິບັດການ



ສະຖາປັດຕະຍະກຳຂອງລະບົບປະຕິບັດການ

- ◆ ລະບົບປະຕິບັດການແບບເຄືອຂ່າຍ ແລະ ກະຈ່າຍວຽກ
 - ລະບົບປະຕິບັດການເຄືອຂ່າຍ
 - ◆ ເຮັດວຽກຢູ່ໃນຄອມພິວເຕີເຄື່ອງໜຶ່ງ, ແຕ່ອະນຸຍາດໃຫ້ຂະບວນ(ໂປຣແກຣມ)ຂອງມັນສາມາດເຂົ້າໄປໃຊ້ຊັບພະຍາກອນໃນຄອມພິວເຕີເຄື່ອງອື່ນໄດ້
 - ລະບົບປະຕິບັດການກະຈ່າຍວຽກ
 - ◆ ເປັນລະບົບປະຕິບັດການອັນໜຶ່ງທີ່ສາມາດບໍລິຫານຈັດການຊັບພະຍາກອນຂອງຫຼາຍເຄື່ອງຄອມພິວເຕີໄດ້
 - ◆ ຈຸດປະສົງ
 - Transparent performance
 - Scalability
 - Fault tolerance
 - Consistency

ສະຖາປັດຕະຍະກຳຂອງລະບົບປະຕິບັດການ



ໜ້າທີ່ຂອງລະບົບປະຕິບັດການ

- ◆ ຄອມພິວເຕີແບບຕັ້ງໂຕະ ແລະ mainframe ຈະມີໜ້າທີ່ແຕກຕ່າງກັນ
- ◆ ໂດຍທົ່ວໄປລະບົບປະຕິບັດການມີໜ້າທີ່ດັ່ງນີ້
 - ດໍາເນີນການແລະ ຄວບຄຸມການເຮັດວຽກຂອງ Application ແລະ ອຸປະກອນຕ່າງໆ
 - ການຈັດສັນຊັບພະຍາກອນທີ່ໃຊ້ຮ່ວມກັນເຊັ່ນ: CPU, RAM, Disk, I/O...

ໜ້າທີ່ຂອງລະບົບປະຕິບັດການ

- ◆ ລະບົບປະຕິບັດການຂອງຄອມພິວເຕີຂະໜາດໃຫຍ່ມີໜ້າທີ່ດັ່ງນີ້:
 - ບໍ່ຫານຈັດການຂະບວນການ (Process Management)
 - ບໍ່ລິຫານຈັດການໜ່ວຍຄວາມຈໍາຫຼັກ (Main Memory Management)
 - ບໍ່ລິຫານຈັດການ File (File Management)
 - ບໍ່ລິຫານຈັດການ I/O (I/O Management)
 - ບໍ່ລິຫານຈັດການ Disk (Disk Management)
 - ບໍ່ລິຫານຈັດການເຄືອຂ່າຍ (Network Management)
 - ປ້ອງກັນລະບົບ (System Protection)
 - ແປບັນດາຄໍາສັ່ງຕ່າງໆ (Command Interpreter)